



*weMeet*

**Treball final de grau**

**Grau en Enginyeria Informàtica**

Disseny i implementació d'una aplicació per a la gestió d'esdeveniments per a la plataforma Android.

# Índex

Índex.....	2
Índex de les figures.....	5
Agraïments .....	7
Introducció .....	8
Motivació.....	8
Estructura del document.....	8
Objectius del treball .....	9
Impacte del projecte .....	9
Innovació del projecte.....	10
Antecedents .....	10
Descripció del problema i característiques.....	10
Anàlisi del mercat.....	11
Resultats de l'anàlisi de mercat.....	14
Desenvolupament del projecte.....	15
Metodologia de desenvolupament.....	15
Anàlisi DAFO .....	16
Planificació .....	17
Planificació temporal prevista.....	19
Diagrama de Gantt .....	20
Desenvolupament del temps real .....	22
Planificació de costos .....	22
Anàlisi de desenvolupament .....	24
Estat dels dispositius intel·ligents .....	24
Android.....	24
iOS .....	25

Windows Phone .....	25
Anàlisi de requeriments .....	26
Requeriments funcionals .....	26
Requeriments no funcionals .....	27
Especificació del sistema.....	28
Diagrames casos d'us .....	28
Model de domini .....	48
Disseny de la solució .....	49
Patrons d'arquitectura utilitzats .....	49
Diagrama de la base de dades online.....	51
Arquitectura del projecte .....	52
Disseny de la interfície de la aplicació.....	53
Implementació .....	55
Tecnologies, programes i eines utilitzades per la implementació .....	55
Android Studio .....	55
Genymotion.....	55
Java Language.....	56
Java Development Kit (JDK).....	56
API de Parse.....	56
API Google Maps .....	56
API Google Places .....	56
eXtensible Markup Language (XML) .....	57
Adobe Illustrator .....	57
Adobe Photoshop.....	57
Implementació de la solució .....	57
Primeres decisions .....	57
Estructura del projecte Android.....	59
Components bàsics aplicació Android.....	60

Implementacions en l'aplicació.....	65
Estructura de l'aplicació .....	65
Integració de la API de Parse.com, base de dades al núvol .....	66
Implementació de la API Parse.com, base de dades al núvol .....	67
Integració Google API's (Maps + Places) .....	69
Implementació Google API's (Maps + Places) .....	73
Implementació Facebook Parse Login.....	75
Implementació de diferents idiomes .....	76
Proves de funcionament .....	76
Proves integració.....	76
Proves de sistema.....	77
Resultats finals .....	78
Splash, accés i creació de usuaris.....	79
Pantalla principal.....	79
Llista d'amics .....	80
Creació d'esdeveniments .....	81
Detall d'esdeveniment i xat.....	82
Conclusions .....	83
Treball futur.....	83
Bibliografia consultada.....	85
Part autodidacta implementació.....	85
Part informativa memòria.....	86

# Índex de les figures

Figura 1- Aplicació Invit -MeetUP.....	11
Figura 2 - Aplicació Meetup- Encuentra tu gente .....	12
Figura 3 - Aplicació We meet together .....	13
Figura 4- Mètode SCRUM.....	16
Figura 5 - Anàlisis DAFO .....	17
Figura 6 - Diagrama de Gantt .....	20
Figura 7- Diagrama de Gantt (Continuació) .....	21
Figura 8 - Taula de costos del projecte.....	23
Figura 9 - Diagrama cas d'ús 01.....	28
Figura 10 - Diagrama cas d'ús 02.....	33
Figura 11 - Diagrama cas d'ús 03.....	43
Figura 12- Model de domini .....	48
Figura 13- Patró d'arquitectura Model Vista Controlador .....	49
Figura 14-Diagrama base de dades .....	51
Figura 15 - Arquitectura del projecte.....	52
Figura 16- Disseny de les interfícies .....	53
Figura 17 - Taules i gràfic versió de Android (2015).....	58
Figura 18- Cicle de vida de l'Activity.....	61
Figura 19- Exemple de vista en el editor de vistes .....	61
Figura 20- Exemple de codi de View .....	61
Figura 21- Comparació entre diferents layouts .....	62
Figura 22- Exemple codi de fragment en View .....	63
Figura 23- Exemple de fragment en visualitzador de View.....	63
Figura 24- Exemple de contingut de Fragment.....	64
Figura 25- Exemple de contingut de fragment en visualitzador de View .....	64
Figura 26- Exemple de Intent explícit.....	65
Figura 27 - Directori de llibreries.....	66
Figura 28- Exemple de codi de inicialització de Parse.....	67
Figura 29- Exemple vista a la web de Parse.com .....	67
Figura 30 - Inserció de objecte a Parse .....	68
Figura 31 - Funcionament de AsyncTask.....	68
Figura 32- Exemple de codi obtenir objectes de Parse .....	69

Figura 33- Dependències del projecte .....	70
Figura 34 - Codi de AndroidManifest.xml .....	70
Figura 35 - Obtenció del certificat.....	71
Figura 36- Creació nova aplicació Google .....	71
Figura 37 - Activació API's Google .....	72
Figura 38- Activació del certificat Google .....	72
Figura 39 - Fragment de codi AndroidManifest.xml .....	73
Figura 40 - Fragment de codi AndroidManifest.xml .....	73
Figura 41 - Codi activació OpenGL .....	74
Figura 42- Codi exemple API mapa de Google .....	74
Figura 43 - Codi exemple API Google Places .....	75
Figura 44 - Codi exemple retorn API Google Places .....	75
Figura 45 - Codi exemple integració de Facebook Parse .....	75
Figura 46 - Directoris diferents idiomes.....	76
Figura 47 - Aplicació a Google Play .....	78
Figura 48 - Splash, accés i creació usuari .....	79
Figura 49 - Pantalla principal.....	80
Figura 50- Opcions d'amics .....	80
Figura 51 - Opcions creació d'esdeveniments.....	81
Figura 52- Opcions creació d'esdeveniments (2) .....	81
Figura 53 - Detalls esdeveniment i xat .....	82

## Agraïments

En primer lloc, voldria agrair el desenvolupament del projecte a tota la meua família, sobretot per aguantar-me en els dies més feixucs de desenvolupament de les parts més difícils del projecte, que són aquells dies que diuen que els que realment t'estimen t'aguanten les males experiències.

En segon lloc, m'agradaria agrair a les persones que realment mai llegiran aquest projecte, però són les persones que expliquen i ensenyen la programació en Android de manera gratuïta en tota la xarxa, i gràcies a aquestes persones i la seva caritat, tots hem podem treure profit i no gastar-nos un dineral en material didàctic.

En tercer i últim lloc, m'agradaria donar un agraïment a la empresa en la que treballo, la qual fa mesos hem va demanar que realitzés una aplicació senzilla per a mòbils, i la experiència hem va agradar tant, que a part de aprendre tot el bàsic per a desenvolupar sobre aquesta plataforma, se'm va il·luminar una bombeta de què es el que realment m'agradava i com se'm passaven les hores volant.

Gràcies.

## Introducció

Avui en dia, la revolució de la telefonia mòbil es un fet, i casi qualsevol persona en un estat desenvolupat disposa d'un telèfon intel·ligent, d'una tableta o d'un televisor també intel·ligent. Aquests dispositius permeten tenir totes les eines de les quals disposa un ordinador a la nostra mà.

Així doncs, tenim un sistema intel·ligent del que gaudeix tothom, i per tant podem assumir que hi ha una connexió massificada globalment.

Aquesta connexió permet un apropament directe a les persones amb les quals es té contacte, i arrel d'aquest fet, poder compartir, elaborar i connectar amb terceres persones. La realització d'aquest projecte, comporta un apropament de les persones, a través del nostre sistema intel·ligent que utilitzem a diari. Es tracta de l'apropament entre persones a nivell social a partir de un aplicació que ho permeti.

## Motivació

Les motivacions per dur a terme aquest projecte han sigut clares des de l'inici de la idea, on els motius importants o personals són el que més ens fa tirar endavant un nou projecte.

El primer d'ells, és crear una aplicació que sigui útil en el dia a dia, que ajudi a la gent, i sobretot que sigui d'ús multitudinari. Tothom imagina com una creació seva, es utilitzada per milers de persones arreu del món.

El segon d'ells, i crec, el més important, és l'aprenentatge d'una nova tecnologia que a dies d'ara està molt sol·licitada en el món laboral, i a partir d'aquest desenvolupament, assolir els coneixements necessaris per poder crear-te un lloc en qualsevol empresa d'aquest àmbit.

Tot això, fa que la feina que es tingui que fer, ja sigui aprenentatge i aplicació dels coneixements, es faci d'una manera més lleugera i amb ganes de fer les coses bé.

## Estructura del document

L'estructura del document és la detallada a continuació:

- Primerament, farem una mica d'introducció, parlant de les motivacions, objectius del treball, l'estat de la matèria que hem triat com a projecte i els impactes que pot tenir el nostre projecte.
- Com a segon capítol, parlarem de les etapes del desenvolupament de l'aplicació, entre les quals destaquen:
  - Metodologia de desenvolupament utilitzada.
  - Planificacions temporals del projecte.
  - Anàlisis DAFO.
  - Anàlisis de requeriments.
  - Casos d'ús.
  - Model de domini de la nostra base de dades.
  - Disseny de l'aplicació.
  - Descripció patrons utilitzats.
  - Arquitectura de la solució.
  - El disseny de les interfícies i de l'aplicació en general.
  - La implementació i les tecnologies utilitzades.



- Com a tercer capítol, parlarem dels resultats obtinguts en base a tot el desenvolupament, i parlarem de l'aplicació final obtinguda, així com les seves possibilitats. També parlarem del treball futur que tenim pensat per la nostra aplicació.
- Per finalitzar, parlarem de les conclusions i mostrarem la biografia consultada.

## Objectius del treball

L'objectiu del projecte es aconseguir fer una aplicació que permeti concretar esdeveniment, cites, aniversaris, o qualsevol tipus de quedada entre un parell de persones o una colla d'amics.

A més a més, el gran objectiu es no tenir que estar pendent de les cites que tens prèvies, o recordar qualsevol data de qualsevol esdeveniment, tot quedarà registra't a l'aplicació, amb els seus conseqüents avisos.

L'aplicació serà distribuïda al Google Play Store, i tindrà les següents característiques:

- Login per usuari, registre de usuari.
- Creació, eliminació i modificació d'esdeveniments.
- Base de dades al Cloud, treballant amb la API de Parse.com.
- Llista d'amics per tal de poder compartir esdeveniments, invitacions.
- GEO localització dels esdeveniments.
- Xat o publicacions entre usuaris a l'esdeveniment.
- Material Design per a tota l'aplicació.
- Sigui capaç de executar-se en la majoria dels dispositius Android del mercat.
- Registri tots els usuaris, llistats d'amics, així com els seus esdeveniments.

Tot això implicarà una gran feina autodidacta en quant parlem del sistema de guardat de dades i recuperació d'aquestes en Cloud, i sobretot, el fer una aplicació pròpia que pugui comportar beneficis.

L'objectiu principal del treball, es agafar coneixement sobre la matèria, programació i desenvolupament en Android, per tal de guiar-me en un futur de desenvolupador d'aplicacions, que es el que realment m'agrada i tinc passió.

## Impacte del projecte

L'impacte que pensem que aquest projecte pot tenir en el mercat és alt. Estem parlant d'una aplicació on fas una connexió d'usuaris amb esdeveniments, tot amb un disseny innovador i desenvolupat perquè tot es faci d'una manera senzilla i pràctica per l'usuari.

A part d'això, l'impacte esperat es que sigui una eina per al ús de cada dia, així doncs, si la seva utilització es diària, obrirà un món molt extens d'usuaris, els quals s'aniran sumant més a més a la nostra aplicació.

## Innovació del projecte

El projecte presenta unes novetats en vers a les demás aplicacions creades i conegudes actualment:

- Disseny cuidat i adaptat al nou model de disseny de Google. S'ha utilitzat Material Design per a cada component de l'aplicació. Tot això aporta una netedat absoluta per la navegació de l'usuari a través de l'aplicació.
- Integració, permet als usuaris tenir esdeveniments personals amb els seus amics, i permet tenir una conversació interna per a cada esdeveniment, per la qual cosa no seran necessàries altres aplicacions de comunicació.
- Permetrà utilitzar la API de Google Place<sup>1</sup>, la versió 2 està pràcticament sortida del forn, i permet localitzar qualsevol establiment conegut en el punt de GEO localització corresponent. A part d'això, ens retorna una sèrie de coordenades, les quals podem utilitzar després per la creació d'un mapa de Google amb aquestes.

## Antecedents

Fins fa pocs anys, els esdeveniments entre amics o entre masses eren part de la creació d'un boca-boca, correus massius informant del que s'esdevindria, o simplement de quedar en un lloc concret en una hora, i arreglar-te-les per arribar al lloc, a l'hora, a cegues de si realment tot seria com s'havia parlat anteriorment.

Amb les noves tecnologies, el tema de la comunicació ha esdevingut un desenvolupament a anys llum del que coneixíem anys enrere. Van emergir nous mètodes de connexió entre gent, ja sigui Facebook<sup>2</sup> o les seves similars xarxes socials que tots coneixem a dies d'ara, i per tant, la comunicació entre les persones ha donat un salt estratosfèric.

Avui en dia el ventall de aplicacions de connexió entre usuaris i xarxes socials s'està començant a saturar, però inevitablement, les que més triomfen són les que permeten donar activitat a altres usuaris d'una manera fàcil i massivament, però com a xarxa social d'esdeveniments, no estan massa desenvolupades ni se'n parla gaire, és una oportunitat excel·lent per poder entrar al mercat i crear una nova xarxa entre amics i coneguts.

## Descripció del problema i característiques

El problema que representa la creació d'aquest projecte, és el disseny d'una aplicació per al sistema operatiu Android, que permeti a l'usuari fer esdeveniments amb els seus amics d'una manera senzilla, guardant les dades pertinents de cada esdeveniment.

---

<sup>1</sup> **Google APIs** : són un conjunt de API's de desenvolupament, creades per Google, que permeten la comunicació amb els serveis de Google i la integració amb altres serveis.

<sup>2</sup> **Facebook**: és una xarxa social llançada el 2004 i que pertany a la companyia privada Facebook, Inc. La xarxa permet afegir gent com a amics, i enviar-los missatges i compartir enllaços, fotografies i vídeos, entre altres coses.

Totes les dades seran guardades en una base de dades online, on l'usuari podrà accedir-hi des de qualsevol lloc on es trobi, i on gràcies a això, podrà intercanviar informació amb els seus amics/invitats dins de cada esdeveniment.

També cal destacar que cada usuari disposarà d'un compte personal, per identificar cada esdeveniment associat, el qual podrà crear a partir de la xarxa social Facebook<sup>3</sup> ràpidament o manualment amb usuari i contrasenya.

### Anàlisi del mercat

Avui en dia, existeixen diferents tipus d'aplicacions de creació d'esdeveniments amb amics, tot això de forma senzilla i dinàmica, i sempre amb mode online, per poder-te assabentar de cada acció en l'esdeveniment a temps real. A continuació, farem un anàlisi de les que més han destacat, i mirarem els seus punts forts i els seus punts febles.

#### Invit- MeetUp Planer

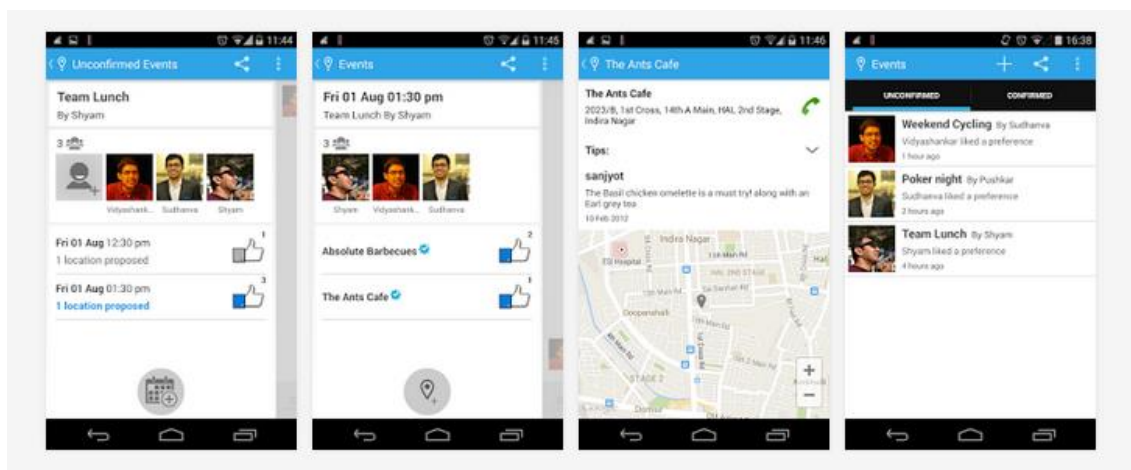


Figura 1- Aplicació Invit -MeetUP

Aplicació d'esdeveniments, desenvolupada per Ushte Studios, que son un grup de desenvolupadors no identificats com a entitat.

L'aplicació permet la creació d'esdeveniments i la invitació dels teus amics a aquests. També permet l'elecció per votació del lloc on es produirà l'esdeveniment. Disposa d'una llista de usuaris confirmats i no confirmats al detall de l'esdeveniment.

<sup>3</sup> **Facebook:** és una xarxa social llançada el 2004 i que pertany a la companyia privada Facebook, Inc. La xarxa permet afegir gent com a amics, i enviar-los missatges i compartir enllaços, fotografies i vídeos, entre altres coses. És obert a tothom qui tingui més de tretze anys, i només cal una adreça de correu electrònic vàlida per registrar-s'hi. És una de les xarxes socials més conegudes actualment.

### Punts forts:

- Disseny net i cuidat
- GEO localització
- Interacció amb diferents usuaris dins de l'aplicació

### Punts febles :

- Encara que el disseny està molt cuidat, no està actualitzat a Material Design<sup>4</sup>.
- No permet la identificació de l'usuari per cap xarxa social exterior
- Els esdeveniments molts cops son confusos, ja que indica la GEO localització de l'esdeveniment, però no podem utilitzar el mapa de Google d'una manera directa a partir de l'aplicació.
- No hi ha comunicació dins de l'esdeveniment, per a possibles malentesos.

### *Meetup- Encuentra tu gente*



*Figura 2 - Aplicació Meetup- Encuentra tu gente*

Aplicació d'esdeveniments multitudinaris creada per MeetUp, societat de la qual no existeix informació. Aquesta aplicació t'informa dels esdeveniments propers de la teva localització. Està enfocada a trobar gent amb les teves mateixes aficions, i així intercanviar experiències.

### Punts forts:

- Aplicació professional , disseny impecable.
- Gran massificació d'esdeveniments.
- Gran connexió entre usuaris dins de l'aplicació.
- Gran ventall de possibilitats per a tots els usuaris, a l'hora d'assistència a esdeveniments.

### Punts febles

- Inexistent en el mercat Europeu, esta enfocada per a un mercat Nord-Americà, falta d'esdeveniments dins d'Europa, on hi ha vegades que no existeix cap.

---

<sup>4</sup> **Material Design:** és l'última tendència de disseny a l'hora de crear aplicacions de Google. (2015)

- Tracte personal entre usuaris inexistent, esdeveniments massius i sense connexió personal.
- Falta de GEO localització de l'esdeveniment.
- Inexistent relació entre usuaris.

### *Wee meet together*



Figura 3 - Aplicació Wee meet together

Aplicació d'obtenció de GEO localització de diferents usuaris. La principal idea que mostra, és l'obtenció de la localització personal i de la persona respectivament connectada simultàniament.

L'aplicació ha sigut desenvolupada per JGUI, que com en les dues anteriors, és una empresa desconeguda de la qual no tenim informació.

### **Punts forts:**

- GEO localització de les dos parts de la connexió entre usuaris.
- Disseny de fàcil enteniment i senzill.

### **Punts febles :**

- Aplicació molt pobre en quant parlem a disseny, ha quedat desfasada per l'usuari d'avui en dia.
- Opcions nul·les o inexistents dins de l'aplicació, sembla un projecte per a gent que comença a programar.

## Resultats de l'anàlisi de mercat

Després d'haver fet la comparació d'aquestes aplicacions, les quals tenen una pinzellada del que volem implementar, ens hem d'assegurar que la nostra aplicació:

- Tingui un disseny innovador i sigui de caràcter professional.
- Tingui una connexió entre usuaris, així com una connexió dins dels esdeveniments.
- Sigui senzilla i fàcil d'utilitzar, de ràpid aprenentatge.
- Els esdeveniments continguin una GEO localització que pugui treballar amb els mapes de Google.

# Desenvolupament del projecte

## Metodologia de desenvolupament

Per a desenvolupar el projecte, s'ha utilitzat una metodologia de desenvolupament àgil, les quals s'utilitzen en qualsevol empresa de desenvolupament de software actuals.

L'emprada per portar a terme aquesta aplicació ha sigut en la mesura possible el mètode Scrum<sup>5</sup>.

El Scrum, es caracteritza pels següents marcs de desenvolupament:

- Adopta una estratègia de desenvolupament incremental, enlloc d'una planificació i execució completa del producte.
- Es basa la qualitat del resultat final en el coneixement de tots els desenvolupadors de l'aplicació.
- Solapament de diferents fases de desenvolupament.

Aquest mètode disposa de diferents rols per ser executat:

- El ScrumMaster: És qui prova de fer seguir les pràctiques de Scrum tant a dins de l'equip com envers l'organització.
- El ProductOwner: És qui representa els interessats en el projecte.
- L'Equip de desenvolupament: Inclou tots els participants en la construcció del programari.

Cada dia del Sprint es realitza una reunió sobre l'estat del projecte, típicament al començament de la jornada. També se li anomena daily standup i té unes regles concretes:

- La reunió comença puntualment a la seva hora. Poden acordar-se càstigs simbòlics - acordats per l'equip- per qui no respecti la puntualitat.
- Tothom és benvingut, tot i que només els membres de l'equip de desenvolupament poden parlar.
- La reunió té una durada fixa de 15 minuts, independentment de la mida de l'equip i de la durada del Sprint.
- Tots els assistents han d'estar dempeus (això ajuda a mantenir la reunió curta)
- La reunió ha de fer-se al mateix lloc i hora durant tot el Sprint.

Durant la reunió, cada membre de l'equip contesta a tres preguntes:

- Que has fet des d'ahir?
- Què faràs avui?
- Quins problemes t'impedeixen arribar als teus objectius del Sprint?

Seguint aquest model de desenvolupament àgil, s'aconsegueix una integració total de tot l'equip envers al producte que s'ha de produir.

---

<sup>5</sup> **Scrum**: sistema de desenvolupament de software àgil.

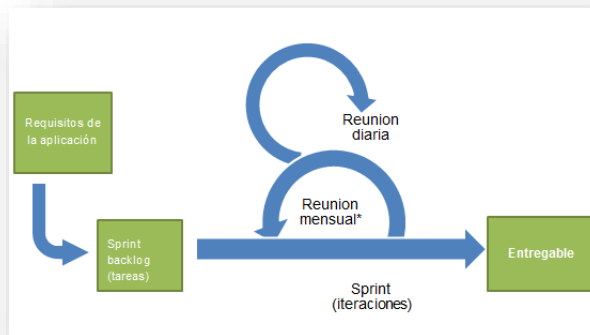


Figura 4- Mètode SCRUM

## Anàlisi DAFO

L'Anàlisi DAFO és un mètode de planificació estratègica per a avaluar les Debilitats, Amenaces, Fortaleses i Oportunitats d'un projecte.

Consisteix en una anàlisi que diferencia entre els factors interns (fortaleses i debilitats) d'una organització i els factors externs d'aquesta (oportunitats i amenaces). Es tracta d'especificar l'objectiu d'un projecte i la identificació dels factors interns i externs que són favorables i desfavorables per assolir aquest objectiu.

- Fortaleses: característiques del projecte que donen un avantatge per assolir els objectius finals del producte.
- Debilitats: desavantatges o limitacions per assolir els objectius.
- Oportunitats: oportunitats externes, del medi, que donen avantatges per millorar els objectius.
- Amenaces : elements externs del medi , que poden causar problemes per assolir l'objectiu.

Hem desenvolupat un anàlisi DAFO per al nostre projecte:



	Punts forts	Punts dèbils
<b>Origen intern</b>	<b>Fortaleses</b> <ul style="list-style-type: none"> <li>-Aplicació innovadora en quant parlem de disseny.</li> <li>-Aplicació gratuïta i lliure de publicitat.</li> </ul>	<b>Debilitats</b> <ul style="list-style-type: none"> <li>-Falta d'experiència a l'hora de crear una aplicació d'Android.</li> <li>-Aplicació només disponible a partir de la versió 4.0 d'Android.</li> <li>-Projecte desenvolupat per una sola persona.</li> </ul>
<b>Origen extern</b>	<b>Oportunitats</b> <ul style="list-style-type: none"> <li>-Necessitat de les persones per utilitzar l'aplicació.</li> <li>- Augment de percentatge de gent que disposa de dispositius Android.</li> <li>- Model de negoci profitós i interès de grans empreses per adaptar el producte als seus grans productes.</li> </ul>	<b>Amenaces</b> <ul style="list-style-type: none"> <li>- Caiguda de la nostra base de dades per part del servei que utilitzem gratuït.</li> <li>- Competència de grans empreses de desenvolupament de Software, copia del nostre producte.</li> <li>- No es creï necessitat d'utilització de la nostra aplicació</li> </ul>

Figura 5 - Anàlisi DAFO

Després d'analitzar els punts forts i febles del nostre projecte, podem resumir que la nostra aplicació pot tenir una gran oportunitat de créixer i ser coneguda, ja que al ser gratuïta i de disseny innovador i amb les opcions que ofereix el desenvolupament mòbil a dies d'ara, tot està a favor de les noves idees i necessitats tecnològiques.

Per altra banda, el tenir que realitzar una aplicació "semi-professional" de manera singular i amb falta d'experiència a l'hora de començar el projecte, son els punts febles més importants. Com comentari final, el que caigui la base de dades d'origen extern es quasi impossible, ja que es una empresa consolidada i comprada per Facebook<sup>6</sup>.

## Planificació

Quan parlem de la planificació, sempre tenim que tenir en compte les nostres necessitats i els nostres deures per obtenir un resultat òptim a l'hora de desenvolupar qualsevol projecte. Es per això, que una bona planificació determina el resultat de tot el desenvolupament, la qual es la part més important del projecte.

A partir de tenir les idees clares del que volíem crear o dur a terme, ens decidim a programar les nostres tasques d'una manera més o menys acceptable per poder treballar a llarg termini, i per això tenim les següents fases de planificació:

<sup>6</sup> **Facebook** : és una xarxa social llançada el 2004 i que pertany a la companyia privada Facebook, Inc. La xarxa permet afegir gent com a amics, i enviar-los missatges i compartir enllaços, fotografies i vídeos, entre altres coses

- Creació de la idea: Primerament, tenim que tenir uns esbossos i una idea clara del que volem fer. Conté consultes amb el professor que et porta el treball, per definir bé les idees que poden sorgir i que s'han d'implementar, i sobretot, tenir clar quins mecanismes i eines utilitzarem per realitzar-la.
- Auto-aprenentatge: Com a segona tasca de procediment del projecte, tenim que aprendre a utilitzar les eines i a conèixer bé tot el món de Android, ja sigui saber utilitzar el espai de treball en el que programarem, aprendre a dissenyar les interfícies de usuari, i fer una pinzella al tema de disseny extern, com poden ser els logos de l'aplicació, les imatges, els vídeos...

Tot això comporta un esforç autodidàctic gran, que és una de les parts més importants a l'hora de afrontar els problemes de creació i resoldre'ls.

- Memòria final: Suposa el redactat de la memòria del projecte, on detallarem tot el relacionat amb el projecte. A això, també li acompanyen les tasques de impressió, enquadernació...
- Primera iteració desenvolupament de l'aplicació: Concretarem un conjunt de requeriments de l'aplicació (entrada d'usuari, creació d'usuari, base de dades, creació de les primeres pantalles d'activitat...), disseny de la base de dades, amb les seves proves corresponents.
- Segona iteració desenvolupament de l'aplicació: Concretarem un conjunt de requeriments de l'aplicació, que en aquest cas seran (creació d'esdeveniment per usuari, mostra de esdeveniments a pantalla principal, ús de les tecnologies de Google, que seran les API's de Places i Maps, comentades anteriorment.) amb les seves proves corresponents.
- Tercera iteració desenvolupament de l'aplicació: Concretarem un conjunt de requeriments de l'aplicació, que en aquest cas seran (detall d'esdeveniment, esdeveniments amb invitats, poder eliminar esdeveniments, usuaris poden afegir a altres usuaris ) amb les seves proves corresponents.
- Quarta iteració desenvolupament de l'aplicació: Concretarem un conjunt de requeriments de l'aplicació, que en aquest cas seran (Possibilitat de comunicar-te dins del detall de l'esdeveniment, creació de usuari mitjançant Facebook) amb les seves proves corresponents.
- Cinquena iteració desenvolupament de l'aplicació: Concretarem un conjunt de requeriments de l'aplicació, que en aquest cas seran per finalitzar el projecte de desenvolupament ( Arreglar la estètica de tota l'aplicació, concretar i arreglar els punts que siguin més febles.) amb les seves proves corresponents.

## Planificació temporal prevista

<i>Planificació del projecte</i>	<i>Temps inicial</i>	<i>Temps real</i>
<i>Inici projecte</i>	2 dies	
<i>Proposta idea</i>	1 dia	1 dia
<i>Matriculació</i>	1 dia	1 dia
<i>Aprentatge eines a utilitzar per al projecte</i>	30 dies	
<i>Desenvolupament Android</i>	15 dies	15 dies
<i>API Bases de dades Web (PARSE)</i>	10 dies	12 dies
<i>Altres API's de interès per a l'aplicació</i>	5 dies	15 dies
<i>Gestió i implementació de la UI</i>	7 dies	
<i>Disseny de la Interface</i>	2 dies	2 dies
<i>Disseny del logo de l'aplicació</i>	2 dies	1 dia
<i>Disseny de la UI del usuari</i>	1 dia	1 dia
<i>Disseny de la UI del esdeveniment</i>	2 dies	1 dia
<i>Implementació registre i login Usuaris</i>	15 dies	
<i>Implementació login</i>	7 dies	9 dies
<i>Implementació registre</i>	2 dies	4 dies
<i>Implementació registre Facebook</i>	7 dies	15+ dies
<i>Implementació Usuari</i>	15 dies	
<i>Implementació llista d'amics</i>	1 dia	3 dies
<i>Implementació esdeveniments per a cada usuari</i>	14 dies	10 dies
<i>Implementació Esdeveniment</i>	15 dies	
<i>Implementació dades dels esdeveniments</i>	1 dia	2 dies
<i>Implementació afegir esdeveniments per usuari</i>	7 dies	7 dies
<i>Implementació GEO localització</i>	2-3 dies	4 dies
<i>Implementació afegir amics als esdeveniments</i>	3-4 dies	8 dies
<i>Implementació interacció usuaris Esdeveniment</i>	15 dies	
<i>Implementació publicacions a esdeveniments</i>	2 dies	10 dies
<i>Implementació assistència esdeveniments</i>	1 dia	No implementat
<i>Implementació afegir amics a l' esdeveniment</i>	1 dia	5 dies
<i>Documentació final de la memòria</i>	20-30 dies	20 dies
<b>TOTAL</b>	<b>+ - 100 dies</b>	<b>+ - 146</b>

A partir d'aquesta planificació temporal, distribuïrem les tasques en un diagrama de Gantt, per poder tenir més ubicades les dates i les seves iteracions de desenvolupament.

## Diagrama de Gantt

				Febrer				Març			
Tasca a realitzar	Data inici	Data final	Total dies	1ª Setmana	2ª Setmana	3ª Setmana	4ª Setmana	1ª Setmana	2ª Setmana	3ª Setmana	4ª Setmana
Inici projecte											
Proposta idea	04/02/2015	04/02/2015	1								
Matriculació	05/02/2015	05/02/2015	1								
Aprenentatge eines a utilitzar per al projecte											
Desenvolupament Android	05/02/2015	21/02/2015	15								
API Bases de dades Web (PARSE)	21/02/2015	28/02/2015	10								
Altres API's de interès per a la App	01/03/2015	05/03/2015	5								
Gestió i implementació de la UI											
Disseny de la interface	09/03/2015	11/03/2015	2								
Disseny del logo de l'aplicació	12/03/2015	15/03/2015	2								
Disseny de la UI del usuari	16/03/2015	17/03/2015	1								
Disseny de la UI del event	18/03/2015	19/03/2015	2								
Implementació registre i login Usuaris											
Implementació login	23/03/2015	29/03/2015	7								
Implementació registre	29/03/2015	31/03/2015	2								
Implementació registre amb Facebook	01/04/2015	07/04/2015	7								
Implementació Usuari											
Implementació llista d'amics	25/04/2015	26/04/2015	1								
Implementació events per a cada usuari	13/04/2015	25/04/2015	14								
Implementació Event											
Implementació dades dels events	27/04/2015	28/04/2015	1								
Implementació afegir events per usuari	02/05/2015	08/05/2015	7								
Implementació geolocalització	11/05/2015	14/05/2015	2								
Implementació afegir amics a l'event	14/05/2015	17/05/2015	4								
Implementació interacció usuaris Event											
Implementació publicacions a events	15/05/2015	17/05/2015	2								
Implementació assistència events	20/05/2015	21/05/2015	1								
Implementació afegir amics a l'event	23/05/2015	24/05/2015	1								
Documentació final de la memòria											
Documentació final de la memòria	01/06/2015	30/06/2015	20-30								

Figura 6 - Diagrama de Gantt

## Diagrama de Gantt (continuació)

				Abril				Maig				Juny			
Tasca a realitzar	Data inici	Data final	Total dies	1ª Setmana	2ª Setmana	3ª Setmana	4ª Setmana	1ª Setmana	2ª Setmana	3ª Setmana	4ª Setmana	1ª Setmana	2ª Setmana	3ª Setmana	4ª Setmana
Inici projecte															
Proposta idea	04/02/2015	04/02/2015	1												
Matriculació	05/02/2015	05/02/2015	1												
Aprenentatge eines a utilitzar per al projecte															
Desenvolupament Android	05/02/2015	21/02/2015	15												
API Bases de dades Web (PARSE)	21/02/2015	28/02/2015	10												
Altres API's de interès per a la App	01/03/2015	05/03/2015	5												
Gestió i implementació de la UI															
Disseny de la interface	09/03/2015	11/03/2015	2												
Disseny del logo de l'aplicació	12/03/2015	15/03/2015	2												
Disseny de la UI del usuari	16/03/2015	17/03/2015	1												
Disseny de la UI del event	18/03/2015	19/03/2015	2												
Implementació registre i login Usuaris															
Implementació login	23/03/2015	29/03/2015	7												
Implementació registre	29/03/2015	31/03/2015	2												
Implementació registre amb Facebook	01/04/2015	07/04/2015	7												
Implementació Usuari															
Implementació llista d'amics	25/04/2015	26/04/2015	1												
Implementació events per a cada usuari	13/04/2015	25/04/2015	14												
Implementació Event															
Implementació dades dels events	27/04/2015	28/04/2015	1												
Implementació afegir events per usuari	02/05/2015	08/05/2015	7												
Implementació geolocalització	11/05/2015	14/05/2015	2												
Implementació afegir amics a l'event	14/05/2015	17/05/2015	4												
Implementació interacció usuaris Event															
Implementació publicacions a events	15/05/2015	17/05/2015	2												
Implementació assistència events	20/05/2015	21/05/2015	1												
Implementació afegir amics a l'event	23/05/2015	24/05/2015	1												
Documentació final de la memòria															
Documentació final de la memòria	01/06/2015	30/06/2015	20-30												

Figura 7- Diagrama de Gantt (Continuació)

## Desenvolupament del temps real

Com era de esperar, no hem disposat de tot el temps que voldríem, i per tant, això s'ha vist afectat en la nostra planificació prèviament acotada.

Molts dels factors que han impedit seguir el esquema temporal, han sigut els següents:

- Falta de temps per horari laboral: Després d'estar programant tot un dia amb l'empresa, les ganes i la actitud per programar a casa eren nul·les, per tant he perdut molts dies entre setmana, cosa que he recuperat els caps de setmana, ja que molts els tenia lliures.
- Complicacions amb iteracions de desenvolupament : Moltes de les coses que hem implementat en l'aplicació son coses innovadores, el qual vol dir que encara que es faci una taula de planificació temporal, les coses es poden complicar i passar dels dos dies que pensaves que estaries desenvolupant el projecte fins a una setmana, per falta de aprenentatge.

Així doncs, algunes de les tasques que primerament havien de portar pocs dies de feina, es van complicar.

Alguns dels casos més concrets son :

- Login amb Facebook
  - Implementació afegir amics als esdeveniments
  - Implementació GEO localització per esdeveniment
- 
- El aprenentatge inicial: Alguns aspectes puntuals de l'aprenentatge, i aspectes més tècnics del desenvolupament que mai havia tocat, han endarrerit molt el temps d'aprenentatge que tenia acotat.

Tots aquests factors han fet que el projecte que inicialment tenia que acabar-se a finals de maig , s'hagi allargat fins a la primera quinzena de juny, cosa inevitable si no tenim prou experiència en el sector ni en els mètodes àgils de desenvolupament.

## Planificació de costos

Farem un anàlisi de pressupost del nostre projecte, incloent tots els recursos materials que hem utilitzat, així com l'esforç humà que hem invertit. Tot plegat, sumarà una xifra que es la que tindríem que tenir en compte si es tractés d'un projecte cap a un client final.

Recursos	Cost
<b>Implementació (hores del treballador)</b> Suposant que un programador junior pot arribar a guanyar una mitja de 60€ diaris, i el projecte ha tingut una durada de 146 dies	<b>8760€</b>
<b>Compta de Google Developer</b> Per poder utilitzar la tenda d'aplicacions de Google, on voldrem inserir la nostra aplicació, haurem de comprar una compta de Developer.	25 \$ = <b>18,90€</b> 1€ =(1,32276 \$)
<b>Equip de desenvolupament</b> Ens referim al equip on treballarem. PC + Monitor i accessoris.	<b>750 €</b>
<b>Dispositius de proves</b> Per provar la nostra aplicació, necessitarem com a mínim un telèfon mòbil i una tableta per tal de assegurar-nos que tot funciona correctament en dispositius reals.	<b>200 € + 250 €</b> Tablet + Mòbil
<b>Despeses varies</b> Aquí incloem tots els gestos d'oficina, així com els imprevistos com el comprar un carregador nou del telèfon de proves, diferents imprevistos amb el equip de proves, etc...	<b>150 €</b>
<b>Cost TOTAL</b>	<b>10128,9 €</b>

Figura 8 - Taula de costos del projecte

Així doncs, tenim que el cost d'aquest projecte rondaria sobre els **10.130€**, cosa més que acceptable per un projecte amb aquestes característiques, i amb la tecnologia que volem utilitzar.

Aquesta seria la factura sense IVA que li passaríem al client.

# Anàlisi de desenvolupament

Després de fer un anàlisi temporal i especificar quines tasques es duran a terme en aquest projecte, definirem quins són els requeriments que ha de complir l'aplicació. Això ho farem definint els casos d'ús per a cada acció amb l'aplicació i descrivint el model de dades.

## Estat dels dispositius intel·ligents

Actualment, el desenvolupament d'aplicacions mòbil es un fet per a qualsevol empresa emergent en el mercat, i per tant, es una oportunitat de futur a hores d'ara que té un gran futur.

A dia d'avui, tenim molts sistemes operatius per als dispositius mòbils. Cadascun d'ells disposa d'unes peculiaritats diferents, i a partir d'aquí, passarem a fer un recull d'aquestes, per als tres principals sistemes operatius més emergents en els dispositius intel·ligents.

## Android

Android és un conjunt de programari per a telèfons mòbils que inclou un sistema operatiu, programari intermediari i aplicacions Google Inc.<sup>7</sup> va comprar el desenvolupador inicial del programari, Android Inc.<sup>8</sup>, el 2005. El sistema operatiu per a mòbils d'Android es basa en una versió modificada del nucli Linux. Android té una quota de mercat del 45% en aquest 2014-2015.

Aquest sistema operatiu disposa de les següents característiques:

- Seguretat : Les aplicacions tenen un determinat nombre de permisos, que el usuari ha d'acceptar abans d'instal·lar en qualsevol dispositiu.
- Codi obert : Plataforma de desenvolupament lliure i de codi obert, centrat en el software lliure.
- Permet gràfics vectorials i animacions basades en Flash, OpenGL...
- Portabilitat : El llenguatge de programació es Java, per tant tenim un gran ventall de execució en diferents tipus de mòbils.
- Gran varietat de serveis incorporats com GEO-localització, sensors, gamepad, termòmetre, acceleròmetre, giroscopi...
- Disposa de multitasca, que això ens permet tenir moltes aplicacions en execució al mateix temps.

El llenguatge de programació utilitzat per desenvolupar aplicacions és Java, llenguatge multi plataforma, interpretat i orientat a objectes, que ofereix un índex de reutilització de codi molt elevat, i en el que existeixen moltes llibreries lliures. La seva plataforma de distribució es el Google Play, on podem trobar tot tipus de aplicacions, jocs, llibres i inclús pel·lícules.

---

<sup>7</sup> **Google Inc:** és l'empresa propietària de la marca Google, el producte principal de la qual és el motor de recerca del mateix nom.

<sup>8</sup> **Android, Inc:** companyia de software ubicada als Estats Units que va desenvolupar Android en els seus inicis, fins que va ser adquirida per Google el juliol de 2005



## iOS

iOS (anteriorment iPhone OS), és un sistema operatiu per a mòbils desenvolupat i distribuït per Apple Inc.<sup>9</sup> Originalment llançat el 2007 per a l'iPhone i l'iPod Touch, s'ha ampliat per suportar altres dispositius d'Apple com l'iPad i l'Apple TV.

A diferència d'altres sistemes operatius, Apple no permet utilitzar el seu sistema operatiu en dispositius que no siguin del mateix fabricant. Avui en dia, disposa d'un 19% de quota de mercat (2014-2015).

iOS deriva de MAC OS X, això indica que té el seu origen en el sistema operatiu UNIX. Apple va anunciar l'existència de iPhone OS el gener de 2007, tot i que fins un any més tard no va tenir el nom oficial iOS.

Les aplicacions han de ser escrites i compilades específicament per l'arquitectura ARM de iOS, conseqüentment, les aplicacions desenvolupades per MAC OS X no poden ser usades en altres plataformes. A partir de març de 2008, es va llançar el Software Development Kit (SDK), per a que desenvolupadors poguessin fer aplicacions per iOS, ja que fins llavors, solament Apple podia desenvolupar aplicacions per aquest sistema operatiu.

A dies d'ara, es poden crear aplicacions per a iOS amb dos llenguatges de programació. El primer es Objective-C, es un llenguatge orientat a objectes amb la fisonomia de C. És un llenguatge dinàmic.

D'altra banda, tenim el llenguatge recentment anunciat que es Swift. Es tracta d'un llenguatge innovador amb una sintaxi més entenedora que el seu antecessor i que produeix aplicacions més lleugeres i ràpides.

La plataforma de distribució d'aplicacions iOS es el Apple Store, on igualment podem trobar aplicacions, jocs, pel·lícules...

## Windows Phone

És un sistema operatiu desenvolupat per Microsoft<sup>10</sup>. Avui en dia disposa d'una quota de mercat entre del 7%, Microsoft va decidir no fer-lo compatible amb Windows Mobile, per tant, les aplicacions existents no serveixen i s'han de tornar a desenvolupar.

El llenguatge de programació utilitzat és C#, llenguatge orientat a objectes desenvolupat i estandarditzat per Microsoft, com a part de la seva plataforma .NET.

La plataforma de distribució d'aquest sistema operatiu es la Windows Phone Store.

---

<sup>9</sup> **Apple Inc:** és una empresa multinacional estatunidenca que dissenya i fabrica productes electrònics de consum i programari per als seus productes.

<sup>10</sup> **Microsoft:** és una empresa multinacional d'informàtica que va ser creada el 21 de setembre de 1975 als Estats Units, on encara hi té la seva seu principal.

## Anàlisi de requeriments

L'anàlisi de requeriments (o requisits) a enginyeria de sistemes i enginyeria de software, comprèn aquelles tasques que determinen les necessitats o condicions que un producte nou o modificat ha de complir per l'usuari final. En el nostre projecte definirem quins són els requeriments funcionals i no funcionals, a partir de les nostres tasques determinades anteriorment en la temporització del projecte.

### Requeriments funcionals

Els requeriments funcionals descriuen quines són les accions que el usuari podrà dur a terme amb la nostra aplicació i què ha de fer la nostra aplicació amb aquesta interacció.

Aquests requeriments seran agrupats per tres principals camps que determinen l'aplicació:

- Esdeveniments
- Usuaris
- Comunicació entre usuaris

#### Usuaris

- (RF 01) - Permet als usuaris registrar-se al sistema, amb usuari i contrasenya.
- (RF 02) - Permet als usuaris accedir a l'aplicació mitjançant les dades del seu registre que han tingut que fer anteriorment.
- (RF 03) - Permet que els usuaris un cop hagin accedit a l'aplicació, poder tancar sessió d'usuari.
- (RF 04) - Permet que l'aplicació guardi l'accés de l'usuari per no tenir que entrar cada cop que iniciem l'aplicació.
- (RF 05) - Permet als usuaris accedir i registrar-se a l'aplicació amb la seva conta de Facebook.

#### Esdeveniments

- (RF 06) - Permet que els usuaris puguin crear un esdeveniment.
- (RF 07) - Permet que els usuaris puguin veure els seus esdeveniments, així com els que han sigut invitats.
- (RF 08) - Permet que els usuaris puguin sortir/eliminar un esdeveniment de la seva llista general d'esdeveniments.
- (RF 09) - Permet que es pugui donar un nom a l'esdeveniment.
- (RF 10) - Permet que es pugui donar una data a l'esdeveniment.
- (RF 11) - Permet que es pugui donar una hora a l'esdeveniment.
- (RF 12) - Permet que es pugui invitar usuaris a l'esdeveniment.
- (RF 13) - Permet que es pugui associar una direcció amb GEO localització a l'esdeveniment.

- (RF 14) - Permet veure el detall de l'esdeveniment, així com les dades de l'esdeveniment, els usuaris que assistiran i els comentaris de cada esdeveniment entre els usuaris assistents.
- (RF 15) - Permet afegir nous usuaris un cop l'esdeveniment ha sigut creat.
- (RF 16) - Permet als usuaris dins del detall de l'esdeveniment, utilitzar el Google Maps a partir de la GEO Localització de l'esdeveniment.

#### Comunicació entre usuaris

- (RF 17) - Permet a l'usuari afegir a usuaris ja registrats a l'aplicació.
- (RF 18) - Permet a l'usuari visualitzar la seva llista d'amics.
- (RF 19) - Permet a l'usuari parlar dins de l'esdeveniment.
- (RF 20) - Permet a l'usuari veure els missatges dels altres usuaris dins de l'esdeveniment.
- (RF 21) - Permet a l'usuari invitar als seus amics dins del detall de l'esdeveniment, un cop creat.

#### Requeriments no funcionals

Els requeriments no funcionals són requisits que expressen restriccions sobre el conjunt de solucions possibles. En altres paraules, són les qualitats generals que ha de tenir l'aplicació a l'hora de desenvolupar-se.

#### Eficiència

- El sistema ha de ser ràpid i aprofitar els recursos del dispositiu.
- Les respostes de acció i reacció de l'aplicació han de ser acceptables.

#### Interfície

- L'aplicació ha de estar feta amb Material Design.
- L'interfície ha de ser senzilla i de fàcil aprenentatge.
- L'aplicació ha de estar neta de publicitat.

#### Fiabilitat

- L'aplicació tindrà un control d'error eficient.
- L'aplicació serà consistent amb qualsevol dispositiu.

#### Seguretat

- L'aplicació només pot ser utilitzada per usuaris registrats.
- Els usuaris només podran accedir als esdeveniments propis o que hagin estat invitats.

#### Portabilitat

- L'aplicació podrà instal·lar-se en un número alt de diferents dispositius/versions.

## Especificació del sistema

L'especificació del sistema consisteix en la descripció de la interacció entre usuari i aplicació. Detallarem l'especificació a partir dels casos d'ús, que definiran amb detall cada acció realitzada, ja sigui prémer un botó, introduir un text... qualsevol cosa relacionada amb la nostra aplicació.

### Diagrames casos d'ús

Els diagrames de casos d'ús identifiquen les funcionalitats en quant parlem de l'usuari envers la nostra aplicació.

Els diagrames han sigut separats idòniament seguint el nostre model de requeriments funcionals. Així doncs, tindrem 3 diagrames d'ús, un per cada apartat dels nostres requeriments.

#### Diagrama cas d'ús Usuaris (Diagrama Cas Ús 01)

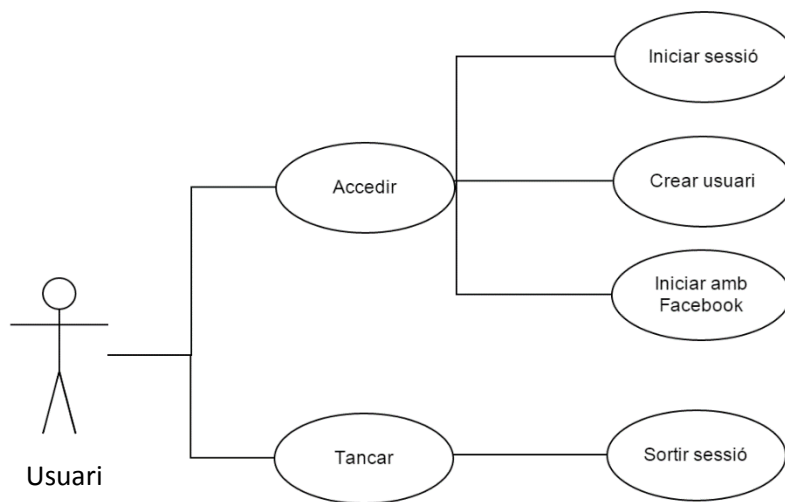


Figura 9 - Diagrama cas d'ús 01

*Descripció cas d'ús Usuaris (Diagrama Cas Ús 01)*

Resum (CU 01)
L'usuari vol entrar a l'aplicació tenint una compta ja creada anteriorment.

Cas d'ús	Entrar	CU 01
Actors	Usuari	
Tipus	Primari, essencial	
Referències	RF 02	
Pre-condició	L'usuari es dins de l'aplicació i vol accedir amb el seus credencials	
Post-condició	L'usuari accedeix a l'aplicació	

Propòsit
Entrar a l'aplicació

Curs típic			
1	L'usuari inicia la aplicació		
2	L'usuari introdueix els credencials		
		3	El sistema comprova que els credencials son correctes
4	L'usuari entra a la pantalla principal d'esdeveniments		

Cursos alternatius	
3	El sistema comprova els credencials i concreta que no son correctes, salta missatge de credencials incorrectes

### Resum (CU 02)

L'usuari vol crear una compta a l'aplicació a partir de la pantalla de registre d'usuari.

Cas d'ús	Entrar	CU 02
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 01	
<b>Pre-condició</b>	L'usuari es dins de l'aplicació i vol crear una compta	
<b>Post-condició</b>	L'usuari crea la compta	

### Propòsit

Crear compta aplicació

### Curs típic

<b>1</b>	L'usuari inicia la aplicació		
<b>2</b>	L'usuari introdueix l'usuari i contrasenya per la creació de la nova compta		
		<b>3</b>	El sistema comprova que els credencials son correctes i que no existeix el usuari introduït
<b>4</b>	L'usuari entra a la pantalla principal de inici de sessió.		

### Cursos alternatius

<b>3a</b>	El sistema comprova els credencials i concreta que el usuari ja existeix, salta missatge d'error de nom d'usuari.
<b>3b</b>	El sistema comprova els credencials i concreta que les contrasenyes introduïdes no coincideixen, salta missatge error de contrasenyes.

<b>Resum (CU 03)</b>
L'usuari vol entrar a l'aplicació tenint una compta de Facebook al dispositiu.

<b>Cas d'ús</b>	<b>Entrar</b>	<b>CU 03</b>
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 05	
<b>Pre-condició</b>	L'usuari es dins de l'aplicació i vol accedir amb el seus credencials de Facebook	
<b>Post-condició</b>	L'usuari accedeix a l'aplicació	

<b>Propòsit</b>
Entrar a l'aplicació a través de Facebook

<b>Curs típic</b>			
<b>1</b>	L'usuari inicia la aplicació		
<b>2</b>	L'usuari prem el botó de accedir mitjançant Facebook.		
		<b>3</b>	El sistema comprova que els credencials son correctes
<b>4</b>	L'usuari entra a la pantalla principal d'esdeveniments		

<b>Cursos alternatius</b>	
<b>3</b>	El dispositiu no disposa de compta Facebook, el inici de sessió es incorrecte.

Resum (CU 04)
L'usuari ha accedit anteriorment a l'aplicació amb una compta d'usuari i vol sortir d'aquesta compta.

Cas d'ús	Entrar	CU 04
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 03	
<b>Pre-condició</b>	L'usuari es dins de l'aplicació i vol tancar la sessió.	
<b>Post-condició</b>	L'usuari surt de la sessió de la compta.	

Propòsit
Tancar la sessió d'usuari

Curs típic			
1	L'usuari inicia la aplicació		
2	L'usuari prem el botó de tancar sessió.		
		3	El sistema tanca sessió i elimina el guardat de sessió de l'usuari.
4	L'usuari entra a la pantalla d'inici de sessió.		

Cursos alternatius
--------------------



Diagrama cas d'ús Esdeveniments (Diagrama Cas Ús 02)

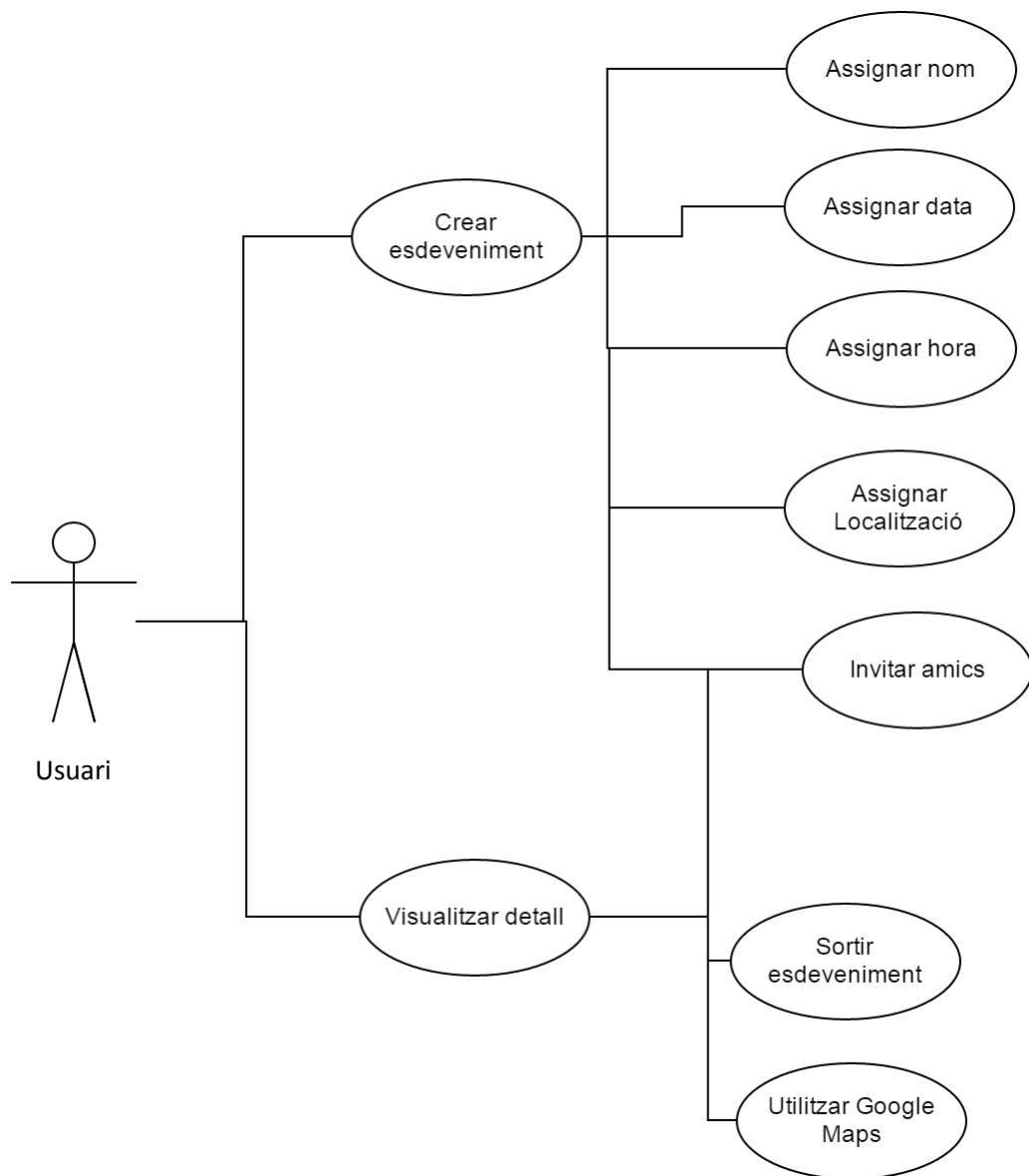


Figura 10 - Diagrama cas d'ús 02

Descripció cas d'ús Usuaris (Diagrama Cas Ús 02)

**Resum (CU 05)**

L'usuari vol crear un esdeveniment

Cas d'ús	Entrar	CU 05
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 06	
<b>Pre-condició</b>	L'usuari es dins de la pantalla principal	
<b>Post-condició</b>	L'usuari crea un esdeveniment	

**Propòsit**

Crear esdeveniment

**Curs típic**

<b>1</b>	L'usuari inicia l'aplicació		
<b>2</b>	L'usuari prem el botó de nou esdeveniment		
<b>3</b>	L'usuari entra totes les dades del esdeveniment		
<b>4</b>	L'usuari prem el botó de "Crear esdeveniment"		
		<b>5</b>	El sistema crea el nou esdeveniment a la base de dades

**Cursos alternatius**

<b>5a</b>	L'usuari no ha plenat tots els camps necessaris per crear l'esdeveniment, no es pot crear l'esdeveniment.
-----------	---

<b>Resum (CU 06)</b>
L'usuari vol assignar un nom a l'esdeveniment

<b>Cas d'ús</b>	<b>Entrar</b>	<b>CU 06</b>
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 09	
<b>Pre-condició</b>	L'usuari es dins de la pantalla de creació d'esdeveniments	
<b>Post-condició</b>	L'usuari assigna un nom a l'esdeveniment	

<b>Propòsit</b>
Assignar nom a l'esdeveniment

<b>Curs típic</b>			
<b>1</b>	L'usuari inicia l'aplicació		
<b>2</b>	L'usuari prem el botó de nou esdeveniment		
<b>3</b>	L'usuari introdueix un nom a l'esdeveniment		

<b>Cursos alternatius</b>
---------------------------

Resum (CU 07)
L'usuari vol assignar una data a l'esdeveniment

Cas d'ús	Entrar	CU 07
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 10	
<b>Pre-condició</b>	L'usuari es dins de la pantalla de creació d'esdeveniments	
<b>Post-condició</b>	L'usuari assigna una data a l'esdeveniment	

Propòsit
Assignar data a l'esdeveniment

Curs típic			
1	L'usuari inicia l'aplicació		
2	L'usuari prem el botó de nou esdeveniment		
3	L'usuari introdueix una data a l'esdeveniment		
		4	El sistema mostra la data seleccionada

Cursos alternatius
--------------------

<b>Resum (CU 08)</b>
L'usuari vol assignar una hora a l'esdeveniment

<b>Cas d'ús</b>	<b>Entrar</b>	<b>CU 08</b>
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 11	
<b>Pre-condició</b>	L'usuari es dins de la pantalla de creació d'esdeveniments	
<b>Post-condició</b>	L'usuari assigna una hora a l'esdeveniment	

<b>Propòsit</b>
Assignar hora a l'esdeveniment

<b>Curs típic</b>			
<b>1</b>	L'usuari inicia l'aplicació		
<b>2</b>	L'usuari prem el botó de nou esdeveniment		
<b>3</b>	L'usuari selecciona una data per a l'esdeveniment		
		<b>4</b>	El sistema mostra la data seleccionada

<b>Cursos alternatius</b>
---------------------------

Resum (CU 09)
L'usuari vol invitar al altres usuaris a l'esdeveniment

Cas d'ús	Entrar	CU 09
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 12	
<b>Pre-condició</b>	L'usuari es dins de la pantalla de creació d'esdeveniments	
<b>Post-condició</b>	L'usuari invita a usuaris a l'esdeveniment	

Propòsit
Invitar usuaris a l'esdeveniment

Curs típic			
1	L'usuari inicia l'aplicació		
2	L'usuari prem el botó de nou esdeveniment		
3	L'usuari inicia la selecció de usuaris amics dintre de l'aplicació		
		4	El sistema mostra la llista d'amics
5	L'usuari selecciona els usuaris que vol invitar a l'esdeveniment		
		6	El sistema mostra la llista d'usuaris invitats per l'esdeveniment

Cursos alternatius	
4	L'usuari no té amics afegits, per tant la llista estarà buida

Resum (CU 10)
L'usuari vol assignar una localització a l'esdeveniment

Cas d'ús	Entrar	CU 10
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 13	
<b>Pre-condició</b>	L'usuari es dins de la pantalla de creació d'esdeveniments	
<b>Post-condició</b>	L'usuari assigna una localització a l'esdeveniment	

Propòsit
Assignar localització a l'esdeveniment

Curs típic			
1	L'usuari inicia l'aplicació		
2	L'usuari prem el botó de nou esdeveniment		
3	L'usuari selecciona el botó de introduir localització		
		4	El sistema mostra un mapa amb la GEO localització actual i permet seleccionar qualsevol establiment conegut.
5	L'usuari selecciona una localització		
		6	El sistema guarda la latitud i longitud de la localització i mostra el lloc seleccionat per l'usuari.

Cursos alternatius	
4	El dispositiu no disposa de Google Maps, l'usuari no podrà seleccionar localització.

<b>Resum (CU 11)</b>
L'usuari vol sortir o eliminar un esdeveniment de la seva compta.

<b>Cas d'ús</b>	<b>Entrar</b>	<b>CU 11</b>
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 08	
<b>Pre-condició</b>	L'usuari es dins de la pantalla de detall de l'esdeveniment	
<b>Post-condició</b>	L'usuari surt/elimina l'esdeveniment	

<b>Propòsit</b>
Sortir/Eliminar esdeveniment

<b>Curs típic</b>			
<b>1</b>	L'usuari inicia l'aplicació		
<b>2</b>	L'usuari selecciona el detall d'esdeveniment		
<b>3</b>	L'usuari pitja el botó de sortir d'esdeveniment		
		4	EL sistema elimina l'usuari de la llista de usuaris que assistiran a l'esdeveniment.

<b>Cursos alternatius</b>
---------------------------



Resum (CU 12)
L'usuari vol invitar a usuaris un cop l'esdeveniment ja ha sigut creat

Cas d'ús	Entrar	CU 12
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 15	
<b>Pre-condició</b>	L'usuari es dins del detall d'esdeveniment	
<b>Post-condició</b>	Els usuaris han sigut invitats a l'esdeveniment	

Propòsit
Afegir usuaris a l'esdeveniment creat

Curs típic			
1	L'usuari inicia l'aplicació		
2	L'usuari selecciona el detall de l'esdeveniment desitjat		
3	L'usuari prem el botó de afegir usuaris		
		4	El sistema mostra la llista d'amics
5	L'usuari selecciona els usuaris que vol invitar a l'esdeveniment		
		6	El sistema mostra la llista d'usuaris invitats per l'esdeveniment

Cursos alternatius	
4	L'usuari no disposa de més amics per seleccionar, o estan tots els seus amics invitats a l'esdeveniment, per tant la llista serà buida.

Resum (CU 13)
L'usuari vol accedir a Google Maps a partir de la GEO localització introduïda

Cas d'ús	Entrar	CU 13
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 16	
<b>Pre-condició</b>	L'usuari es dins del detall d'esdeveniment	
<b>Post-condició</b>	S'inicia Google Maps amb la localització de l'esdeveniment	

Propòsit
Utilitzar Google Maps a partir de la GEO localització del esdeveniment.

Curs típic			
1	L'usuari inicia l'aplicació		
2	L'usuari selecciona el detall de l'esdeveniment desitjat		
3	L'usuari prem el botó de marca d'esdeveniment al mapa		
		4	El sistema mostra les opcions del mapa
5	L'usuari selecciona la icona de la ruta.		
		6	El sistema inicia Google Maps amb la GEO localització seleccionada.

Cursos alternatius	
6	El dispositiu no disposa de Google Maps, l'usuari no podrà seleccionar localització.

Diagrama cas d'ús Esdeveniments (Diagrama Cas Ús 03)

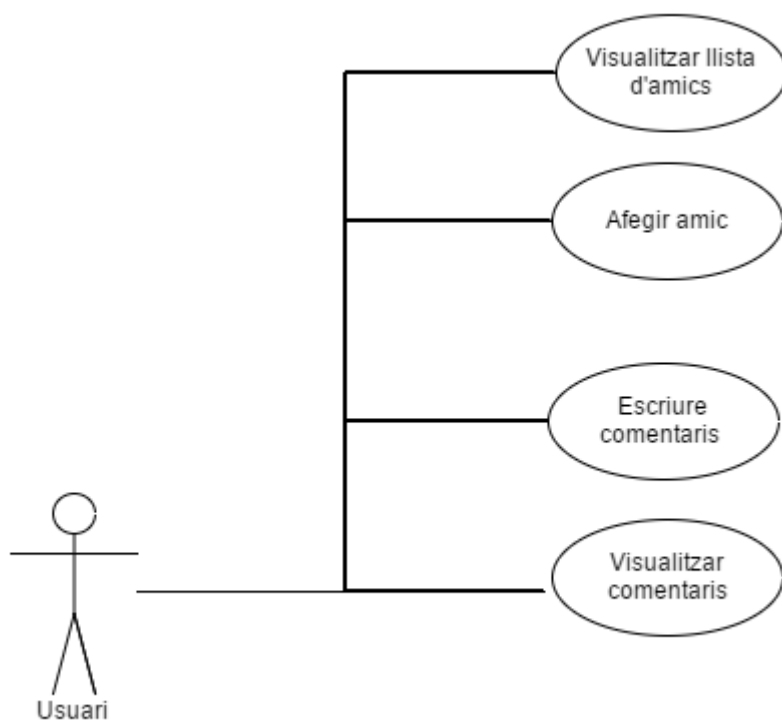


Figura 11 - Diagrama cas d'ús 03

Descripció cas d'us Usuaris (Diagrama Cas Ús 03)

**Resum (CU 14)**

L'usuari vol afegir a un altre usuari prèviament registrat a l'aplicació.

Cas d'ús	Entrar	CU 14
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 17	
<b>Pre-condició</b>	L'usuari es dins de la llista d'amics	
<b>Post-condició</b>	L'usuari afegeix al seu amic usuari2	

**Propòsit**

Afegir nou amic usuari.

**Curs típic**

<b>1</b>	L'usuari inicia l'aplicació		
<b>2</b>	L'usuari selecciona la llista d'amics		
<b>3</b>	L'usuari prem el botó de afegir nou usuari.		
		4	El sistema mostra una pantalla per afegir l'usuari.
<b>5</b>	L'usuari escriu el nom de l'usuari2.		
		6	El sistema afegeix a l'usuari2 i el mostra a la taula de amistats de l'usuari principal

**Cursos alternatius**

<b>6</b>	L'usuari2 no existeix al sistema, per tant no pot ser afegit.
----------	---

<b>Resum (CU 15)</b>
L'usuari vol visualitzar la llista d'amics afegits.

<b>Cas d'ús</b>	<b>Entrar</b>	<b>CU 15</b>
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 18	
<b>Pre-condició</b>	L'usuari es dins de l'aplicació	
<b>Post-condició</b>	L'usuari visualitza la llista d'usuaris afegits com a amics	

<b>Propòsit</b>
Visualitzar llista d'amics.

<b>Curs típic</b>			
<b>1</b>	L'usuari inicia l'aplicació		
<b>2</b>	L'usuari selecciona la llista d'amics		
		<b>3</b>	El sistema mostra una llista d'amics.

<b>Cursos alternatius</b>	
<b>3</b>	L'usuari no disposa de cap amic encara, per tant la llista serà buida.

<b>Resum (CU 16)</b>
L'usuari vol veure els comentaris dins de l'esdeveniment.

<b>Cas d'ús</b>	<b>Entrar</b>	<b>CU 16</b>
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 21	
<b>Pre-condició</b>	L'usuari es dins de l'esdeveniment	
<b>Post-condició</b>	L'usuari visualitza els comentaris	

<b>Propòsit</b>
Visualitzar els comentaris d'altres usuaris dins de l'esdeveniment.

<b>Curs típic</b>			
<b>1</b>	L'usuari inicia l'aplicació		
<b>2</b>	L'usuari selecciona detall d'esdeveniment		
<b>3</b>	L'usuari prem el botó de comentaris		
		<b>4</b>	El sistema mostra la llista de comentaris.

<b>Cursos alternatius</b>
---------------------------

<b>Resum (CU 17)</b>
L'usuari vol comentar coses dins de l'esdeveniment amb altres usuaris invitats.

<b>Cas d'ús</b>	<b>Entrar</b>	<b>CU 17</b>
<b>Actors</b>	Usuari	
<b>Tipus</b>	Primari, essencial	
<b>Referències</b>	RF 22	
<b>Pre-condició</b>	L'usuari es dins de l'esdeveniment	
<b>Post-condició</b>	L'usuari visualitza i comenta dins del detall d'esdeveiment	

<b>Propòsit</b>
Parlar amb altres usuaris dins de l'esdeveniment.

<b>Curs típic</b>			
<b>1</b>	L'usuari inicia l'aplicació		
<b>2</b>	L'usuari selecciona detall d'esdeveniment		
<b>3</b>	L'usuari prem el botó de comentaris		
<b>4</b>	L'usuari introdueix un comentari		
		4	El sistema mostra el comentari escrit en la pantalla de comentaris.

<b>Cursos alternatius</b>
---------------------------

## Model de domini

Un model de domini es un model conceptual de tots els temes relacionats amb un tema específic. Dins d'aquest model de domini, es descriuran totes les seves entitats, atributs, papers, relacions entre tots els components.

A grosso modo, el model de domini proporciona una visió estructural del domini que pot ser complimentat pel model dels casos de ús. Defineix un vocabulari i es útil a la hora de representar el problema.

Així doncs, hem creat un model de domini per a resoldre el nostre problema :

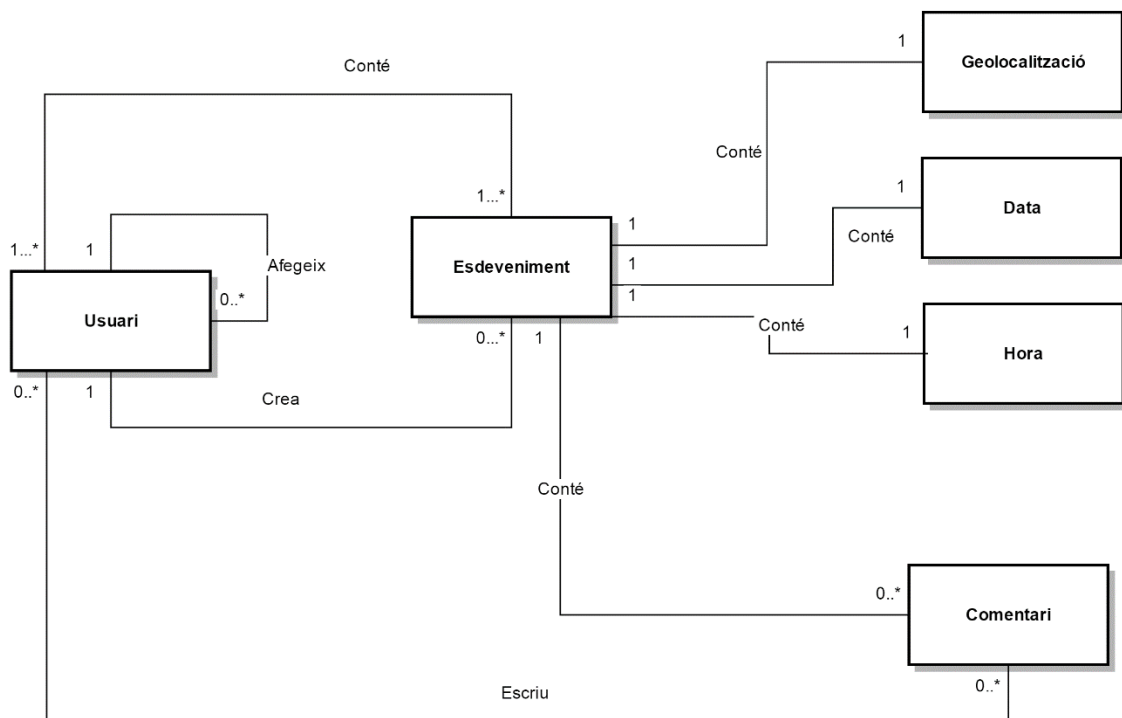


Figura 12- Model de domini

En aquest model de domini podem observar 2 entitats importants, les quals són :

- **Usuari:** Un usuari pot afegir a altres usuaris, així com crear un esdeveniment i escriure comentaris dins dels esdeveniments.
- **Esdeveniment:** Un esdeveniment es creat per un usuari, el qual conté 1 o molts usuaris. Un esdeveniment també conté una GEO localització, una data i una hora. A part de tot això, un esdeveniment també conté cap o molts comentaris escrits per qualsevol usuari que pertany a l'esdeveniment.

A partir d'aquí, ja tenim totes les relacions muntades i per tant ja podem seguir amb la resolució del nostre problema.



## Disseny de la solució

Després d'haver fet un anàlisi de requeriments i els seus models de domini i casos d'ús, ens disposarem a descriure de quin disseny i components estarà disposada la nostra aplicació.

Per fer tot això, descriurem els patrons d'arquitectura utilitzada per resoldre el problema, definirem el diagrama de la nostra base de dades online, el diagrama d'arquitectura de la solució i finalment mostrarem el disseny de la interfície per tenir un esquema de la connexió entre tots els elements.

### Patrons d'arquitectura utilitzats

#### Model-Vista-Controlador

Aquest patró de desenvolupament de programari divideix l'aplicació en tres parts interconnectades :

- **Model:** És el model de dades de la nostra solució, el qual disposa de les arquitectures i solucions dels objectes que farem servir per emmagatzemar dades i fer-les servir. En el nostre cas, Android, seran les classes per defecte.
- **Vista:** És la part que l'usuari veu per pantalla i la que li permet tenir una visió de l'aplicació a l'usuari. En Android, aquesta vista es representa en format XML<sup>11</sup> i conté tots els elements de la nostra interfície de usuari.
- **Controlador:** És l'element que permet fer la interacció entre les accions de l'usuari sobre la nostra aplicació, poden manipular el model de dades i actualitzen la vista segons les accions que es duguin a terme.

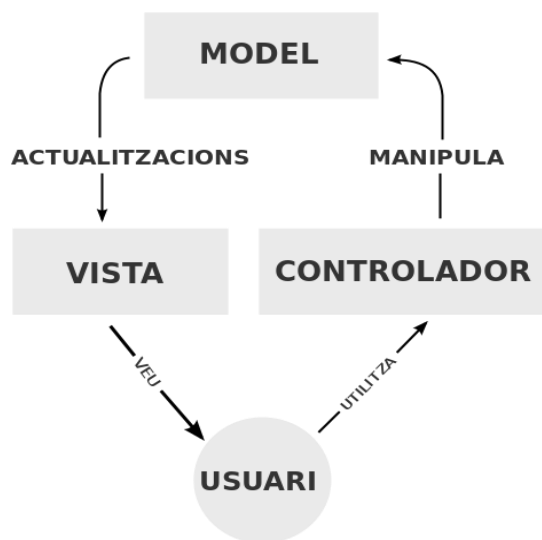


Figura 13- Patró d'arquitectura Model Vista Controlador

<sup>11</sup> **XML:** és un metallenguatge extensible, d'etiquetes, desenvolupat pel World Wide Web Consortium (W3C).

En la nostra aplicació, tenim els següents elements per a cada part del patró de desenvolupament MVC (Model-View-Controller):

<b>Model (Classes)</b>	<b>Vista (Vistes en XML)</b>	<b>Controlador (Activitats)</b>
Comment	Vista_addFriend.xml	AddFriendActivity
Event	Vista_addFriendToEvent.xml	AddFriendToEventActivity
User	Vista_addFriendToEventCreated.xml	AddFriendsToEventCreatedActivity
	Vista_comments.xml	CommentsActivity
	Vista_createEvent.xml	CreateEventActivity
	Vista_detailEvent.xml	DetailEventActivity
	Vista_friends.xml	FriendsActivity
	Vista_login.xml	LoginActivity
	Vista_mainEvents.xml	MainEventsActivity
	Vista_setUsername.xml	setUsernameActivity
	Vista_splash.xml	SplashActivity
	Vista_userRegistration.xml	UserRegistrationActivity

## Diagrama de la base de dades online

Per resoldre el problema de l'emmagatzemat de dades, ens hem decantat per fer-ho de manera online via la API de Parse<sup>12</sup>, la qual ens proporciona una interfície visual via Web, i una sèrie de objectes que ens deixen interactuar amb les nostres dades via Java, fent consultes, inserint objectes o dades i recuperant-los en qualsevol cas. Més endavant entrarem en més detall amb aquesta API i la seva tecnologia, així com els seus procediments per fer-la servir.

Aquí doncs, escenificarem el nostre model de base de dades utilitzat per desenvolupar la nostra aplicació. El nostre model conté 4 taules principals:

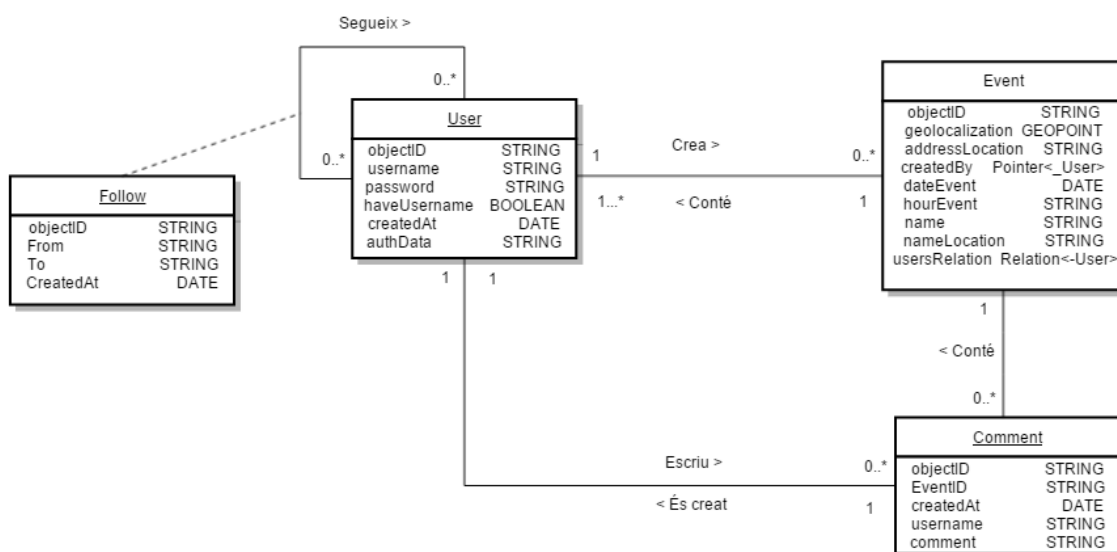


Figura 14-Diagrama base de dades

- **User**: Taula on afegirem tots els usuaris un cop registrats, ja sigui per via normal de registre o per via Facebook. Per fer-ho de la segona manera, utilitzarem el camp **haveUsername**, ja que quan es fa la petició a Facebook per utilitzar els credencials de la mateixa aplicació, ens crea un username aleatori, i per tant, per poder contactar amb els altres usuaris, sempre que iniciem la aplicació amb Facebook per primer cop, es demana que s'indiqui quin nom d'usuari es vol.
- **Follow**: Taula on es guardarà el registre de amistats dels usuaris. Cada cop que un usuari afegeix a un altre a la seva llista d'amics, es registra com **From** el nom del primer usuari, i com a **To** el nom del usuari afegit. Així doncs podem fer una consulta ràpida per obtenir tots els amics que un usuari **From** té sobre la columna **To**.

<sup>12</sup> **Parse**: Es una API que ens permet tenir una connexió directa entre la nostra aplicació i la nostra base de dades online, proporcionada per ells. Parse es una plataforma que permet tenir una base de dades al núvol de forma gratuïta, i utilitzar les seves funcions pre-definides per tal de tenir una interacció directa.

- Event: Aquesta es la taula més gran que tenim, i la que conté més relacions entre objectes. Primerament, tenim una relació **Pointer<\_User>** la qual apunta al usuari que ha creat l'esdeveniment. Així, si en un futur volem tocar temes d'administració d'esdeveniments, donaríem tots els poder al usuari que ha creat l'esdeveniment.

Per altra banda, tenim la **Relation<\_User>**, que és una relació amb la taula de Usuaris, i permet emmagatzemar tots els usuaris existents en aquesta condició. En altres paraules, es una sub-taula que conté tots els usuaris invitats a l'esdeveniment.

- Comment: La taula de comentaris, està relacionada amb un esdeveniment directe, amb el eventId, i per cada comentari, conté el nom d'usuari pel qual ha sigut creat.

### Arquitectura del projecte

L'arquitectura del projecte pretén mostrar d'una manera clara i entenedora, amb un esquema de tecnologies i actors, quin és el funcionament de la nostra aplicació a temps real.

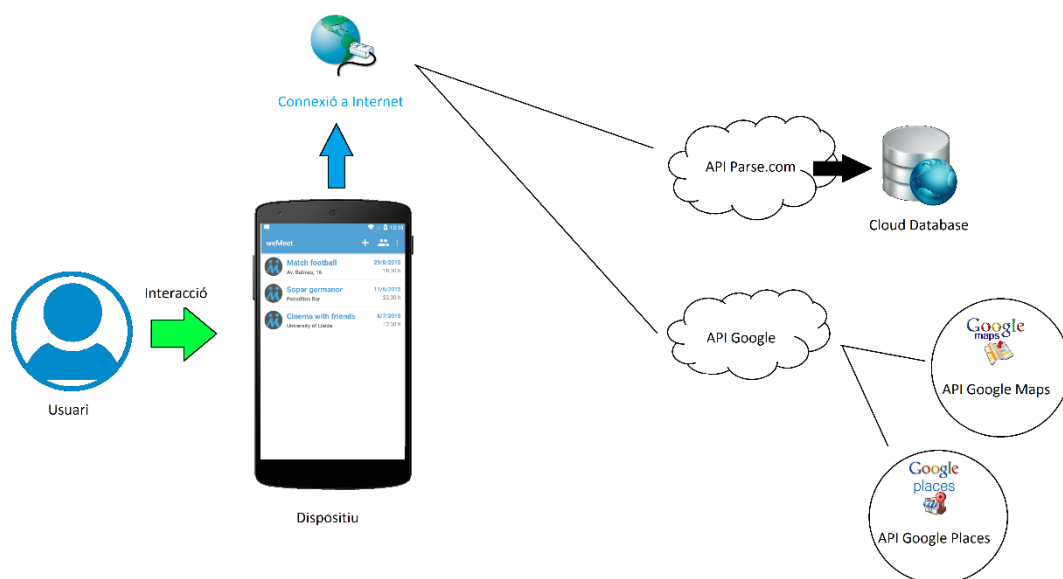


Figura 15 - Arquitectura del projecte

En aquest esquema d'arquitectura del projecte, podem observar els diferents objectes que fan la interacció entre usuari-aplicació-xarxa.

Primerament tenim el Usuari, que a partir de un controlador del nostre projecte, realitza una acció sobre la aplicació. Un cop tenim la acció de l'usuari, i sempre que aquesta necessiti fer una consulta a través de internet, es connectarà a la xarxa i farà la consulta pertinent a la API que volem utilitzar.

Si es tracta de una consulta cap a la API de Parse.com, aquesta interactuarà amb la nostra base de dades ubicada al núvol, on tenim tota la informació guardada de la nostra aplicació per a cada usuari.

Si pel contrari es tracta de una consulta a nivell de la API de Google, serà perquè l'usuari està creant un esdeveniment i vol seleccionar una GEO localització a partir de la API de Google Places<sup>13</sup>, o bé perquè un cop l'esdeveniment ha sigut creat, l'usuari se'n va a l detall d'esdeveniment i allí pot visualitzar en un mapa i utilitzant la API de Google Maps el lloc exacte de la localització triat a partir de la creació del esdeveniment.

## Disseny de la interfície de la aplicació

Un cop hem explicat totes les parts de disseny intern de la nostra aplicació, passarem a comentar el disseny d'usuari, fent un esquema de les activitats de la nostra aplicació, les quals són les pantalles que l'usuari veu de forma real al seu dispositiu mòbil, i amb les quals pot interactuar. En aquest cas, com hem fet referència abans, en el patró de Model-Vista-Controlador, parlarem de la Vista.

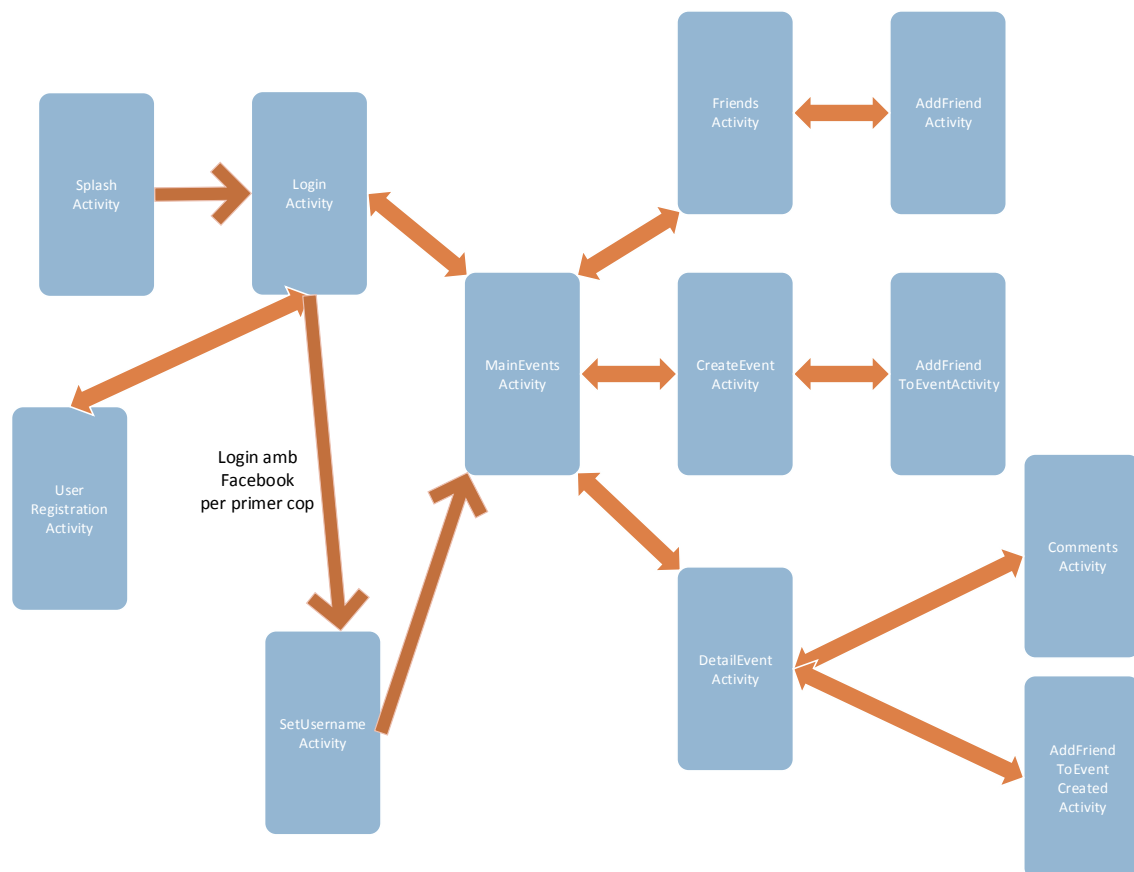


Figura 16- Disseny de les interfícies

En aquest disseny podem veure les següents activitats, que són els nodes dels components de l'aplicació, els quals serveixen per la interacció usuari-aplicació:

<sup>13</sup> **Google Places:** API que ens permet seleccionar qualsevol establiment conegut per Google a partir d'un mapa que ens facilita aquesta mateixa.

- **Splash Activity** : Mostra la càrrega de la aplicació amb una icona i un component de espera de càrrega.
- **Login Activity**: Mostra la pantalla per accedir a l'aplicació, així com el botó per registrar-se o per entrar a l'aplicació amb Facebook.
- **UserRegistration Activity**: Mostra una vista de un registre de usuari, al finalitzar correctament el registre, ens retorna a *Login Activity*.
- **SetUsername Activity**: Mostra una pantalla per si es el primer cop que entrem amb els credencials de Facebook. Ens permet assignar un nom d'usuari al registre.
- **MainEvents Activity**: És la pantalla principal de l'aplicació. Mostra els esdeveniments que hem creat o que hem sigut invitats. També mostra els botons de anar a la llista d'amics, de afegir un nou esdeveniment o de entrar al detall de un esdeveniment. També tenim la opció de tancar sessió de usuari i tornar al *Login Activity*.
  - **Friends Activity**: Mostra la llista d'usuaris que hem afegit com a amics. També conté un botó per poder afegir un nou usuari com a amic.
    - **AddFriend Activity**: Mostra una pantalla per poder introduir un nom d'usuari per afegir-lo. Si afegim un usuari, automàticament ens redirigeix a *FriendsActivity*.
  - **CreateEvent Activity**: Mostra una pantalla per poder crear un esdeveniment. Li podem donar un nom, una data, una hora, una localització i invitar als nostres amics.
    - **AddFriendToEvent Activity**: Mostra una pantalla per poder seleccionar els amics que volem afegir.
  - **DetailEvent Activity**: Mostra el detall d'un esdeveniment ja creat. A partir d'aquesta pantalla, podem sortir del esdeveniment, podem veure els comentaris de l'esdeveniment i fins i tot podem afegir nous usuaris a l'esdeveniment.
    - **Comments Activity**: Aquesta vista ens permet comentar coses a l'esdeveniment i interactuar amb els altres usuaris que han estat invitats.
    - **AddFriendToEvent Created Activity**: Aquesta pantalla ens permet afegir a nous usuaris a l'esdeveniment un cop ja ha sigut creat.

## Implementació

Com a última fase de procés de desenvolupament de l'aplicació, passarem a la part d'explicació de la implementació.

Aquesta disposarà en primer lloc de les tecnologies utilitzades per tal de poder desenvolupar l'aplicació, en segon lloc, parlarem del desenvolupament de la implementació, de la estructura de un projecte de Android, dels components bàsics utilitzats, de l'estructura de l'aplicació i de les implementacions de les API's de l'aplicació, que en aquest cas seran Google Maps, Google Places i la API de Parse.

Per finalitzar, definirem les proves que han sigut realitzades per tal de verificar que l'aplicació complia tot el desitjat.

### Tecnologies, programes i eines utilitzades per la implementació

Per tal de crear una aplicació amb Android, hem utilitzat els següents programes i eines per tenir una experiència de desenvolupament profitós i que ens permeti treballar d'una manera eficient i alegre.

Així doncs, presentarem els programes , tecnologies i eines utilitzades per poder dur a terme aquesta aplicació.

#### Android Studio

Android Studio es un entorn de desenvolupament integrat (IDE) per la plataforma de Android. Va ser anunciat per el 16 de maig de 2013. És un software gratuït i està basat amb un model de desenvolupament única i exclusivament de Android. Està disponible per Windows<sup>14</sup>, Mac OS<sup>15</sup> X i Linux<sup>16</sup>.

A dies d'ara es el entorn més innovador que tot desenvolupador d'aplicacions Android utilitza. Així doncs, ha sigut la nostra principal eina de desenvolupament del nostre projecte.

És un software totalment enfocat al desenvolupament de Android, el qual et dona totes les opcions gràfiques i tecnològiques per tal de poder treballar amb totes les opcions del mercat, a l'hora de fer una aplicació.

#### Genymotion

És un emulador de Android que permet crear repliques virtuals dels sistemes de Android i fer-les córrer en el sistema operatiu de Windows. Aquesta eina es externa al programa de *Android Studio* però pot ser afegida com a extensió i poder-la utilitzar com a dispositiu per provar el nostre codi en fases de desenvolupament.

Com a característica principal, es molt més ràpid que l'emulador que et proporciona *Android Studio* i per tant els temps de càrrega disminueixen molt notablement.

---

<sup>14</sup> **Microsoft Windows:** és una sèrie de sistemes operatius interfícies gràfiques d'usuari produïts per Microsoft.

<sup>15</sup> **El Mac OS:** (Macintosh Operating System) és un sistema operatiu creat per Apple Computer ('ordinador poma' en anglès) per a ordinadors Apple Macintosh.

<sup>16</sup> **Linux:** denominat per alguns GNU/Linux, és el sistema operatiu format pel nucli o kernel Linux, juntament amb les utilitats GNU.

## Java Language

Java es un llenguatge de programació de propòsit general, concurrent, orientat a objectes i va ser específicament dissenyat per tenir poques dependències de implementació com fos possible.

La característica principal es que es portable, ja que permet als desenvolupadors d'aplicacions que escriguin un programa i a partir d'aquí poder-lo executar en qualsevol dispositiu.

És el llenguatge que hem utilitzat per escriure gran part de la nostra aplicació, i és el llenguatge natiu de Android.

## Java Development Kit (JDK)

És un software que ens prové unes eines de desenvolupament per la creació de programes amb *Java*. Pot ser instal·lat en una computadora local o en una unitat de xarxa.

Ens aporta un compilador de *Java*, un intèrpret, genera vistes prèvies de les applets<sup>17</sup>, entre d'altres.

## API de Parse

L'API de Parse es una solució completa de back-end per a la nostra aplicació de dispositiu mòbil. Ens proporciona uns determinats serveis al núvol, com en el nostre cas ha sigut la base de dades.

Lo que realment fa atractiu utilitzar la API de Parse es la simplicitat amb la que pots interactuar amb simples objectes que et proporciona la mateixa API, per tal de poder fer consultes i obtenir respostes de la teva base de dades.

És gratuïta fins a un cert número de consultes i respostes per segon, i per segons quantes persones estan utilitzant la seva API amb una mateixa aplicació.

A part de tot això, ens permet integrar Facebook a la nostra aplicació, ja que té una connexió directa amb la API de Facebook utilitzant els objectes i les consultes de Parse.

## API Google Maps

La API de Google Maps ens permet utilitzar els serveis de Google Maps i integrar-los en el nostre dispositiu, a la nostra aplicació.

Google Maps es un servidor d'aplicacions de mapes a la web que pertany a la empresa Google. Ofereix imatges de mapes desplaçables, així com fotografies per satèl·lit de tot el món.

## API Google Places

La API de Google Places, com la de *Google Maps*, ens permet utilitzar els serveis de Google Places i integrar-los en el nostre dispositiu i aplicació.

---

<sup>17</sup> **Un applet:** es un component d'una aplicació que s'executa en el context d'un altre programa, per exemple en un navegador web.



Google Places és una subvarietat de Google Maps, la qual disposa de totes les localitzacions com comerços, tendes, hospitals, llocs en concret localitzables. Ens permet adquirir aquests llocs i saber la seva informació, així com la seva GEO Localització.

### eXtensible Markup Language (XML)

És un metallenguatge extensible, d'etiquetes, desenvolupat pel World Wide Web Consortium (W3C)<sup>18</sup>. És un estàndard per a l'intercanvi d'informació estructurada entre diferents plataformes.

És el llenguatge que hem utilitzat per poder programar les nostres Vistes (Interfície de Usuari) a la nostra aplicació Android.

### Adobe Illustrator

És un editor de gràfics vectorials desenvolupat per Adobe Systems Incorporated<sup>19</sup> en forma de taller d'art que treballa sobre un tauler de dibuix, i està destinat a la creació artística de dibuix i pintura de il·lustració.

És una de les eines que hem utilitzat per fer la part gràfica del logotip de l'aplicació, així com algunes icones.

### Adobe Photoshop

És un editor de gràfics desenvolupat per Adobe Systems Incorporated. Utilitzat principalment per al retoc de fotografies i gràfics, el seu nom vol dir taller de fotografia.

L'hem utilitzat junt amb el *Adobe Illustrator* per poder fer el logotip i icones de diferents mides, per tal de poder-se adaptar a qualsevol dispositiu mòbil.

## Implementació de la solució

Un cop tenim tota la part teòrica i tota la informació recopilada per tal de fer abast a tot el projecte, ens iniciem al desenvolupament.

Primerament hem de decidir en quina plataforma ho farem, i en aquest cas, per la experiència en el camp i per la facilitat a l'hora de fer testeig i programar ens decantem fer-ho a la plataforma Android.

Ja hem parlat de les característiques de Android, però una de les més importants per triar-la també es que es software lliure, per tant els costos addicionals de desenvolupament són nuls.

### Primeres decisions

Tot seguit, abans de començar qualsevol projecte en aquesta plataforma, hem de determinar un parell de coses que ens acompanyaran al llarg de tot el viatge de desenvolupament.

---

<sup>18</sup> **El World Wide Web Consortium** : (W3C) és un consorci internacional que treballa per a desenvolupar i promoció estàndards per al World Wide Web.

<sup>19</sup> **Adobe Systems Incorporated** : és una empresa de programari amb seu a San José (Califòrnia, EUA) fundada el desembre de 1982 per John Warnock i Charles Geschke.

### Versió de API

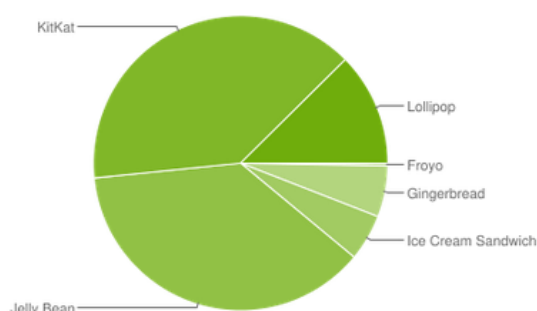
La primera d'elles serà determinar quina serà la versió de API mínima de Android en la qual el nostre projecte serà executat.

En el nostre cas, la versió mínima de API serà la 15, que en versió de Android correspon a la 4.0.3 - 4.0.4. Hem triat aquesta mínima versió, ja que a dies d'ara, el **90,4%** dels dispositius Android arreu del món podran utilitzar-la.

Un altre motiu pel que hem triat aquesta versió es per el disseny Material Design. En dispositius amb versions anteriors a la API 15, alguns dels components visuals que utilitzarem no disposen de les eines necessàries per ser executats, i per tant obtindríem una aplicació plena de errors i es faria casi impossible adaptar-la per a tots els dispositius coneguts.

Com a versió base a l'hora de la implementació, hem utilitzat la versió de API 21, que correspon a Android Lollipop , versió 5.0, i que és la més nova fins a la data. Ens permet un ventall molt gran de possibilitats visuals i és a la qual tots els dispositius nous s'estan o estaran actualitzats.

Version	Codename	API	Distribution
2.2	Froyo	8	0.3%
2.3.3 - 2.3.7	Gingerbread	10	5.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5.1%
4.1.x	Jelly Bean	16	14.7%
4.2.x		17	17.5%
4.3		18	5.2%
4.4	KitKat	19	39.2%
5.0	Lollipop	21	11.6%
5.1		22	0.8%



Data collected during a 7-day period ending on June 1, 2015.

Any versions with less than 0.1% distribution are not shown.

Figura 17 - Taules i gràfic versió de Android (2015)

En aquest gràfic podem apreciar a dia 1 de juny de 2015 l'estat de percentatge per API-dispositius. Podem veure que hem fet una bona elecció, ja que només se'ns escapa la quota de mercat d'un 5,9%, que pertany a les versions Froyo (2.2 - 0,3%) i Gingerbread (2.3.3,2.3.7 - 5,6%).

### Permisos de l'aplicació

Per altra banda, la nostra aplicació també requerirà d'una sèrie de permisos per tal de poder utilitzar tots els components que componen l'aplicació. Aquests permisos simplement són per avisar a l'usuari quines funcionalitats farà servir l'aplicació del telèfon, com poden ser el GPS, la càmera de fotos, l'agenda, la xarxa d'internet....

Tots aquests permisos es troben en un arxiu del projecte que té el nom de AndroidManifest.xml, el qual és un fitxer que determina tota la configuració de la aplicació, així com la funcionalitat de les activitats de l'aplicació.

En la nostra aplicació hem trobat necessaris els següents permisos:

```
android:name="android.permission.WRITE_EXTERNAL_STORAGE"
android:name="android.permission.ACCESS_COARSE_LOCATION"
android:name="android.permission.ACCESS_FINE_LOCATION"
android:name="com.google.android.providers.gsf.permission.READ_GSERVIC
ES"
```

Els permisos llistats a dalt determinen l'ús de la GEO localització en el nostre dispositiu, cosa totalment necessària per crear un esdeveniment en la nostra aplicació, i per veure el detall d'esdeveniment en el mapa de Google Maps.

```
android:name="android.permission.INTERNET"
android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Els següents permisos, són referents a l'ús de internet en el dispositiu, que els necessitem per tal de poder accedir a la nostra informació que tenim en la base de dades al núvol.

Sense els permisos llistats a dalt, la aplicació no podria tenir un bon ús, i per tant, com bé hem comentat abans, l'aplicació no pot ser instal·lada abans d'acceptar-los.

## Estructura del projecte Android

Just abans de comentar la implementació de la nostra aplicació i els seus components, descriurem com s'organitzen els arxius en un projecte de Android, així com les classes de Java, els models de dades, les activitats, els components, les vistes, els arxius d'idiomes i els arxius de imatges.

- Carpeta amb el nostre nom de aplicació, en el nostre cas, weMeet:
  - Conté les carpetes amb els components de tota l'aplicació, i a l'arrel, conté l'arxiu build.gradle, que es on s'especifica les versions de Android que s'utilitzaran com a mínima i màxima per tal de poder executar-se, i les llibreries que agafarem de internet en forma de paquet per poder compilar la nostra aplicació.
- Directori **build** :
  - Conté tots els arxius generats a partir d'una compilació de la nostra aplicació.
- Directori **libs**:
  - Conté les llibreries importades amb extensions .jar.
- Directori **Res**:
  - Conté els recursos utilitzats per l'aplicació. Aquesta carpeta conté subcarpetes per cada tipus de recurs.
- Directori **src**:

- Conté tot el relacionat amb el codi de l'aplicació. A l'arrel d'aquest directori, podem trobar el arxiu AndroidManifest.xml, és un fitxer que determina tota la configuració de la aplicació i ja hem comentat abans.
- Subdirectori **java**:
  - Conté totes les classes i activitats de la nostra aplicació.
- Subdirectori **res**:
  - Conté tots els recursos utilitzats en la nostra aplicació
    - **Drawable** : icones i imatges.
    - **Layout** : vistes de cada activitat.
    - **Menu** : opcions del menú opcions de cada activitat.
    - **Values**: conté els arxius de valors de colors, de dimensions, de estils i de strings que hem utilitzat com a estàtics per a tota la nostra aplicació.

## Components bàsics aplicació Android

De manera que ja sabem com funciona l'estructura d'un projecte Android, passarem a descriure quins són els components bàsic de qualsevol aplicació Android, per tal de saber com es comporta la nostra aplicació i el funcionament dels seus components.

### Activity

Una activitat, o activity, representa el component principal de la interfície gràfica d'una aplicació. Són les pantalles de l'aplicació, i la seva funció bàsica és crear la interfície d'usuari. Una activitat té un cicle de vida, que en aquest cas podem controlar des de la nostra aplicació, però en cas de que el sistema necessiti recursos i la nostra activitat els estigui utilitzant, el mateix sistema operatiu pot destruir-la i quedar-se amb aquests recursos.

Qualsevol activitat disposa d'un cicle de vida, que té els següents estats:

- **onCreate()**: Es crida en la creació de l'activitat. Normalment, s'utilitza per les inicialitzacions, per la creació de la interfície d'usuari i per la inicialització d'estructures de dades.
- **onStart()**: Es crida quan l'activitat està a punt de ser mostrada a l'usuari.
- **onResume()**: Es crida quan l'activitat començarà a interactuar amb l'usuari.
- **onPause()**: Es crida quan l'activitat està a punt de ser enviada a segon pla. És el lloc adequat per emmagatzemar dades que s'estan editant
- **onStop()**: Es crida quan l'activitat està a punt de deixar de ser visible per l'usuari
- **onRestart()**: Es crida quan l'activitat està a punt de tornar a ser mostrada a l'usuari, després d'haver passat per onStop()
- **onDestroy()**: Es crida quan l'activitat està a punt de ser totalment destruïda. És possible que aquest mètode no sigui cridat, en el cas que hi hagi molt poca memòria disponible

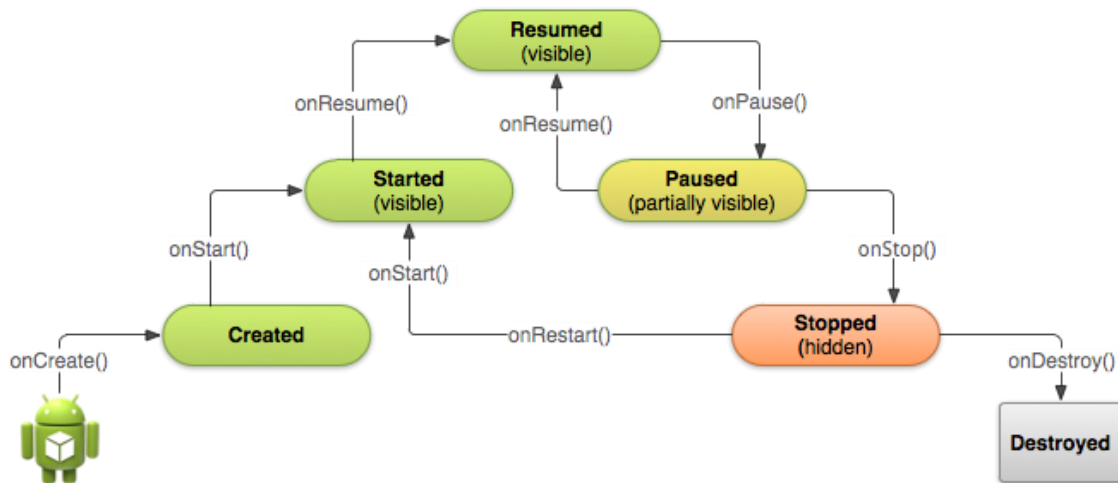


Figura 18- Cicle de vida de l'Activity

## View

Les vistes de la nostra aplicació definiran els components de cada activitat, i és la part visual la qual l'usuari veurà a la seva pantalla. Es defineixen a partir d'un arxiu XML, i son carregades a partir de cada activitat en el moment de la creació, per tant, en el model explicat abans, **són creades en el mètode onCreate()**.

Disposem de una varietat enorme de components per poder jugar en qualsevol vista, que poden ser, formats de text plans, introducció de textos, botons, calendari, hora, imatges, botons de verificació, taules o llistes de informació, fragments, i un seguit de components que fan que la nostra aplicació disposi de tot el necessari per poder ser implementada al nostre gust.

```

<RelativeLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true">

    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="150dp"
        android:id="@+id/LayoutImage"
        android:layout_marginTop="20dp">

        <ImageView
            android:layout_width="150dp"
            android:layout_height="150dp"
            android:layout_alignParentTop="true"
            android:layout_centerHorizontal="true"
            android:src="@drawable/logo_splash"/>

        </RelativeLayout>

    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_below="@+id/LayoutImage">

```

Figura 20- Exemple de codi de View

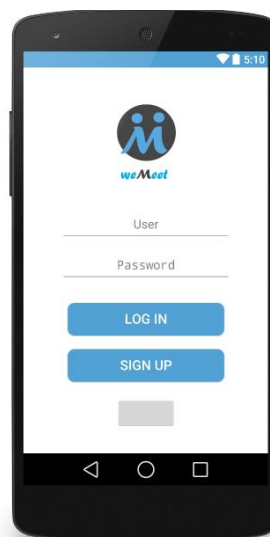


Figura 19- Exemple de vista en el editor de vistes

## Layout

Els layouts son uns components que fan de pautes i pares de altres components. Majoritàriament, serveixen per tal de poder controlar els elements de la vista dins de la nostra *Vista*. Són elements no visibles per l'usuari, i marquen els marges i espaiat entre els diferents components que la componen.

Podem diferenciar dos tipus de Layouts:

- **RelativeLayout:** Els components interiors a aquest tipus de Layout es distribueixen de forma relativa, i per tant, no segueixen cap ordre estricte de posicionament.
- **LinearLayout:** Els components que la componen segueixen una norma estricta de marges dins d'aquest element, ja que van un seguit darrere de l'altre. En aquest cas, podem parlar de forma horitzontal o vertical, i la mida del component Layout definirà la mida màxima i mínima dels components que la completen.

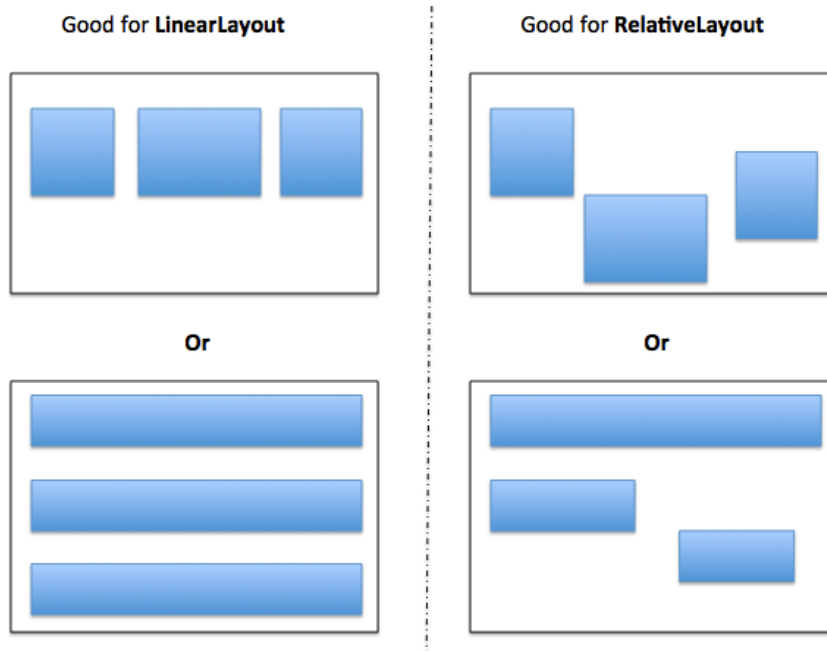


Figura 21- Comparació entre diferents layouts

## Fragment

Un Fragment representa un comportament o una porció d'interfície d'usuari en una activitat. Es poden combinar diversos fragments en una sola activitat per construir una interfície d'usuari multi - panell i reutilitzar un fragment en múltiples activitats .Es pot pensar en un fragment com una secció modular d'una activitat , que té el seu cicle de vida , rep els seus propis esdeveniments d'entrada , i que es poden afegir o treure mentre que l'activitat està en marxa.

Els fragment són afegits a la vista de l'activitat on els volem utilitzar, i ells mateixos tenen la seva pròpia vista, la qual és independent del codi de vista de l'activitat.

```
<fragment
    android:layout_marginTop="5dp"
    android:layout_width="fill_parent"
    android:layout_height="150dp"
    android:name="smallelephant.wemeet.Fragments.UsersEventFragment"
    android:id="@+id/fragmentUsers">

</fragment>
```

Figura 22- Exemple codi de fragment en View

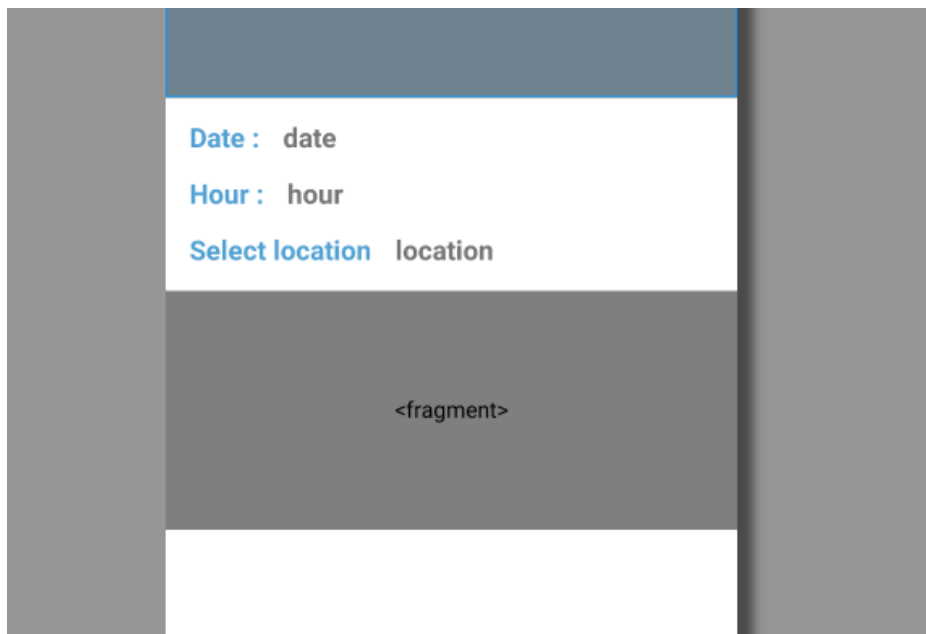


Figura 23- Exemple de fragment en visualitzador de View

Aquí podem observar un exemple de activitat que conté un fragment, el qual conté una llista de objectes, representats en un ListView, que és una representació de una llista d'objectes.

Més a baix, tenim la implementació de la vista del fragment en sí, que es la que es carrega al inserir el fragment en l'activitat.

```
tools:context="smallelephant.wemeet.Fragments.UsersEventFragment">

<ListView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/ListViewUsersFragment"
    android:dividerHeight="0sp"
    android:background="#EEEEEE">
</ListView>
```

Figura 24- Exemple de contingut de Fragment

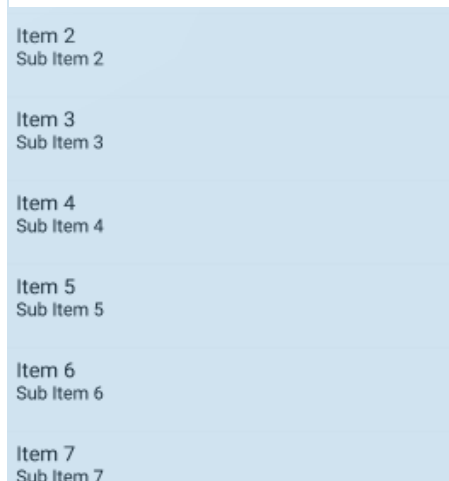


Figura 25- Exemple de contingut de fragment en visualitzador de View

D'aquesta manera, podem tenir un conjunt de components dintre de una activitat que funcionen per ells mateixos, i que per tant, l'activitat només recrea i no s'ha de vincular més amb el cicle de vida d'aquets objectes.

### Intent

Per finalitzar les nostres explicacions dels components bàsics, parlarem dels Intents. Una Intenció és un objecte de missatgeria, que proporciona una execució d'unió entre el codi en diferents aplicacions en l'entorn de desenvolupament d'Android.

Per tant, es un intent de fer qualsevol acció, ja sigui obrir una nova activitat o passar dades entre diferents activitats o components, ja siguin fragments, activitats o diferents components visuals.

Existeixen dos tipus diferents de intents:

- **Intents explícits:** Aquets tipus de intents, com ja indica la paraula que el descriu, són intents encarats a una cosa en concret, com pot ser donar inici a una nova activitat.

#### Exemple:

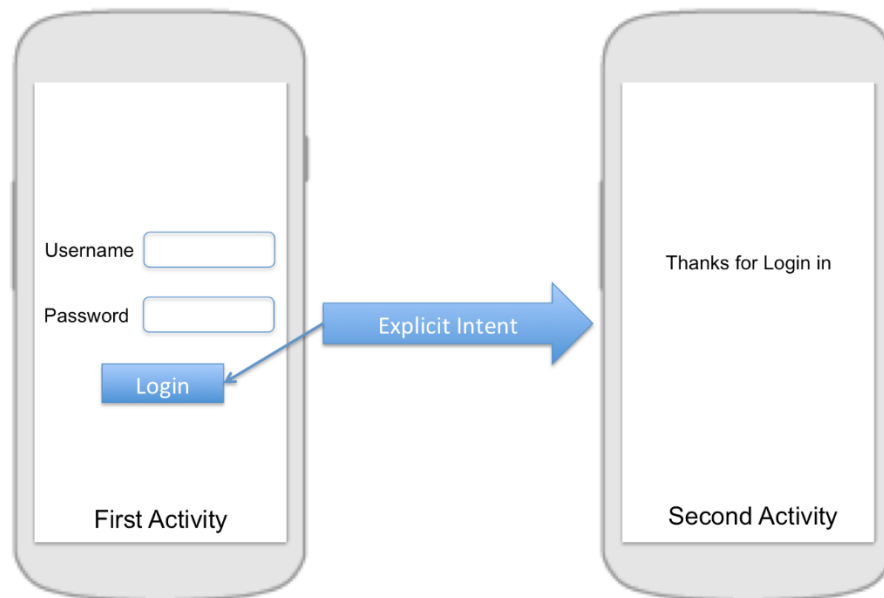
```
Intent(LoginActivity.class,MainActivity.class);
```

- **Intents implícits:** no especifiquen directament els components Android que haurien de ser cridats, només especifica l'acció que s'ha de realitzar.



**Exemple:**

```
Intent(ACTION_VIEW,Uri.parse("http://www.google.com"));
```



*Figura 26- Exemple de Intent explícit*

## Implementacions en l'aplicació

Per finalitzar la part de desenvolupament de la implementació, parlarem de les implementacions de codi de la nostra aplicació, que seran la implementació de tot el tema relacionat amb el funcionament de la nostra base de dades al núvol, així com les diferents integracions de les API's de Google. També parlarem de la implementació per tal de tenir la aplicació multi idioma.

### Estructura de l'aplicació

La estructura i implementació de l'aplicació es diferencia en dues parts. La primera es l'aplicació que s'executa al mòbil, els elements que la conformen i els seus components, que podríem dir que es la part local.

L'altra implementació es la que conté les accions contra la base de dades, guardat de dades, obtenció de dades, creació de relacions... i a part de la base de dades, la part de connexió amb les API's de Google per tal de poder jugar amb la GEO localització i els llocs d'interès.

Així doncs, podem definir la nostra implementació en la part de codi pur de la nostra aplicació, i la part de joc amb les API's corresponents a la xarxa.

L'aplicació disposa de usuaris, que son guardats en la nostra base de dades al núvol. Cada usuari s'ha de registrar per poder utilitzar la nostra aplicació. Ho pot fer de manera "manual", definint un nom de usuari i una contrasenya, o pel contrari i de manera més còmoda, registrar-se via Facebook per tal de saltar-se el pas del registre. Si el usuari es registra en mode Facebook, obligatòriament deurà introduir un nom d'usuari per a que els demás usuaris de l'aplicació el puguin afegir.

La part dels esdeveniments també està allotjada en la nostra base de dades al núvol. Qualsevol usuari, al crear un nou esdeveniment, aquest guarda tota la informació i dades a la base de dades, i els altres usuaris, poden veure aquest nou objecte si pertanyen amb ell.

A part, cada esdeveniment pot contenir uns comentaris, que no son més que un text associat a un usuari i a un esdeveniment. Cada usuari que pertany a l'esdeveniment pot consultar aquests comentaris, i pot afegir-ne de nous.

Com podem apreciar, tot el tema de dades està allotjat a la nostra base de dades del núvol, i a continuació explicarem i detallarem com funciona aquesta base de dades, a l'hora de fer la creació d'aquesta mateixa, crear nous objectes que pertanyin a taules ja creades a través de la web, i a jugar amb les dades que ja tenim guardades.

### Integració de la API de Parse.com, base de dades al núvol

Com ja hem comentat abans, Parse.com es una sistema web que ens permet utilitzar el conjunt de tecnologies que ofereix aquesta empresa. En aquest cas, nosaltres hem utilitzat el sistema de base de dades remot al núvol.

Per poder utilitzar aquestes funcionalitats, el primer que tenim que fer es registrar-nos a la web de **Parse.com** i crear-nos una nova aplicació de Android, que contindrà les diferents tecnologies que en ofereixen, entre elles, el sistema de base de dades.

Un cop ens hem registrat i hem creat la nostra aplicació, tenim que baixar-nos la última llibreria de Parse.jar i integrar-la en el nostre projecte.

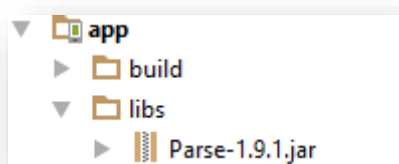


Figura 27 - Directori de llibries

Després de tot això, el nostre projecte farà una auto-compilació de la llibreria i per tant ja podrem utilitzar totes les funcionalitats que ens ofereix aquesta API.

Per últim, tenim que crear al nostre projecte una classe que ens permeti identificar la inicialització de la nostra API amb la nostra aplicació creada a **Parse.com**. Ells mateixos ens ofereixen dos valors, que son les claus que tindrem que escriure a l'hora de inicialitzar el component de la API.

```

public class ParseApplicationClass extends Application {

    @Override
    public void onCreate() {
        super.onCreate();

        Parse.enableLocalDatastore(getApplicationContext());
        Parse.initialize(this, "ANUSVING", "ANUSVING");
    }
}

```

Figura 28- Exemple de codi de inicialització de Parse

## Implementació de la API Parse.com, base de dades al núvol

Un cop tenim integrada la API de parse, ja podem utilitzar totes les seves funcions i mètodes que ens permeten tenir una connexió directa amb la nostra base de dades, la qual podem visualitzar a través de la web.

Data		+ Row	- Row	+ Col	Security	More
Role	0	<input type="checkbox"/>	objectId String	createdAt Date	updatedAt Date	
User	38	<input type="checkbox"/>	5fwVggngf8	Jun 27, 2015, 08:24	Jun 27, 2015	
Comment	82	<input type="checkbox"/>	iKjyKHqWjV	Jun 22, 2015, 19:39	Jun 22, 2015	
Event	36	<input type="checkbox"/>	00VctMTktX	Jun 22, 2015, 19:12	Jun 22, 2015	
Follow	57	<input type="checkbox"/>	F4zXAdRBp7	Jun 22, 2015, 19:11	Jun 22, 2015	
		<input type="checkbox"/>	6n4D48Ieml	Jun 22, 2015, 19:06	Jun 22, 2015	
		<input type="checkbox"/>	xFlQPP4nE6	Jun 20, 2015, 15:31	Jun 22, 2015	
		<input type="checkbox"/>	nz5T9RyZCm	Jun 20, 2015, 09:27	Jun 20, 2015	
		<input type="checkbox"/>	m0V9cumxWt	Jun 18, 2015, 14:41	Jun 18, 2015	

Figura 29- Exemple vista a la web de Parse.com

Primerament parlarem de com inserir objectes en la nostra base de dades, que es la part més important per començar a utilitzar la nostra API de forma correcta.

```
ParseObject gameScore = new ParseObject("GameScore");
gameScore.put("score", 1337);
gameScore.put("playerName", "Sean Plott");
gameScore.put("cheatMode", false);
gameScore.saveInBackground();
```

Figura 30 - Inserció de objecte a Parse

El que fem en aquest tros de codi es inserir un nou objecte, en aquest cas GameScore. Indiquem les variables de les quals disposarà aquest objecte, i tot seguit, guardem el objecte amb la funció **saveInBackground()**.

Aquesta funció auto-crida a una funció de **AsyncTask**, que no es més que una funció en un altre fil diferent del fil principal de l'activitat, així aconseguim que l'aplicació no es quedi penjada mentre interactua amb la nostra base de dades en el servidor remot de *Parse*.

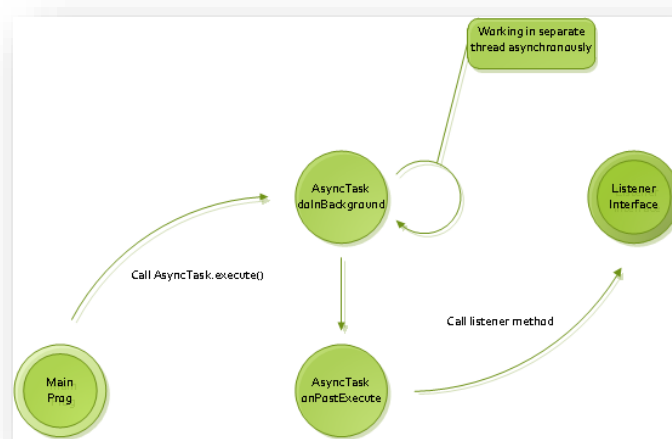


Figura 31 - Funcionament de AsyncTask

Si el que volem es recuperar un objecte de la nostra base de dades, hem de utilitzar la funció **findInBackground()**, que utilitza el mateix mètode AsyncTask que per guardar un objecte.

```

ParseQuery<ParseObject> Query = ParseQuery.getQuery("Follow");
Query.whereEqualTo("user", ParseUser.getCurrentUser().getUsername().toString());
Query.findInBackground((new FindCallback<ParseObject>() {
    @Override
    public void done(List<ParseObject> friendsListParse, ParseException e) {

        if (e==null){
            for(ParseObject parseObject : friendsListParse){
                String nameFriend =(String) parseObject.get("name");
                friendsList.add(nameFriend);
            }
            mListView.setAdapter(new rowAdapter());
        }
        else {
            Toast.makeText(getActivity(), "Can not connect to the DataBase",
                Toast.LENGTH_LONG).show();
        }
    }
})).

```

Figura 32- Exemple de codi obtenir objectes de Parse

En aquesta tros de codi, podem veure com recuperem els usuaris de la taula "Follow", i li passem com a Query que una de les nostres variables sigui igual al usuari que ha accedit a la nostra aplicació, i que es guarda automàticament a través de la nostra API.

Un cop s'executa la funció, ens retorna una llista de objectes ParseObject que recuperem les seves dades, en aquest cas el nom, i els emmagatzemem en una llista de Strings.

Un cop tenim la llista de amics que ens ha retornat la nostra base de dades, la afegim a la nostra vista, en aquest cas mListView, que es el component de llista que tenim en la nostra activitat.

## Integració Google API's (Maps + Places)

Per integrar les API's de Google en la nostra aplicació i poder utilitzar els serveis que aquestes ens proporcionen , el primer que tenim que fer es configurar el nostre projecte per tal de poder tenir una vinculació amb els serveis de Google.

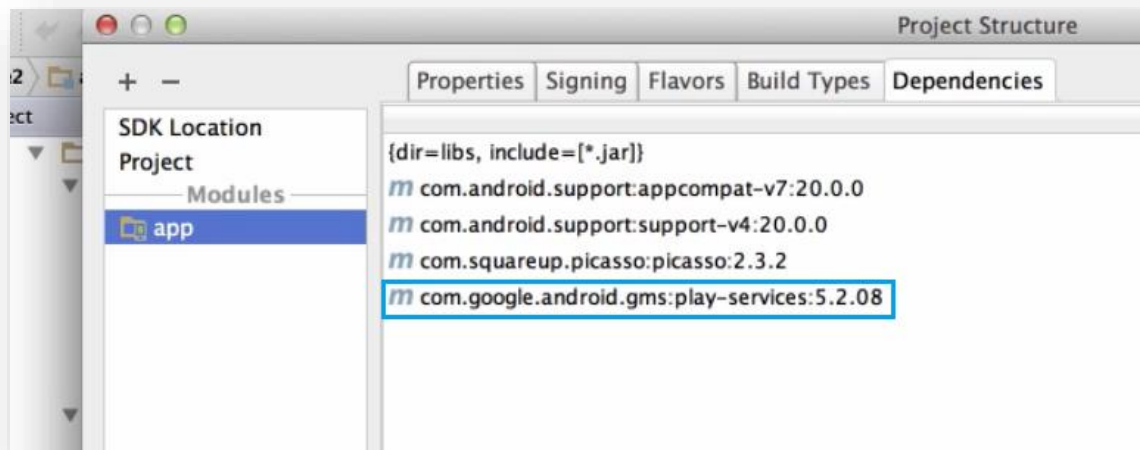


Figura 33- Dependències del projecte

- El primer es afegir la dependència marcada a la imatge de dalt, per tal de poder utilitzar els serveis de Google.
- Després d'això, el següent pas es indicar al nostre arxiu AndroidManifest.xml que l'aplicació utilitzarà aquestes dependències, i ho farem de la següent manera :

```

<meta-data android:name="com.google.android.gms.version"
    android:value="5208" />

</application>

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

</manifest>

```

Figura 34 - Codi de AndroidManifest.xml

- Seguint amb els passos, haurem de crear un certificat digital per tal de poder utilitzar els serveis de Google, i el que farem es executar aquest codi en la consola de Windows, en aquest cas:

---

```
keytool -list -v -keystore "%USERPROFILE%\android\debug.keystore" -alias
androiddebugkey -storepass android -keypass android
```

---

Això ens generarà una clau SHA1, que es la que utilitzarem per registrar la aplicació als serveis de Google i poder utilitzar les seves funcionalitats.

```
valid from: Tue Sep 16 18:56:37 BST 2014 until: Thu Sep 08 18:56:37 BST 2044
Certificate fingerprints:
  MD5:  50:44:7D:7A:21:C0:4E:BA:8C:36:C7:63:08:52:5E:FE
  SHA1: 4F:9E:25:1F:49:45:B4:74:62:28:25:39:88:3F:03:90:75:19:F5:4C
  SHA256: 78:91:42:58:D1:46:84:E6:F6:25:BA:8D:BD:CE:BB:EC:48:0C:D7:FD:93:
0F:17:C9:8F:70:C3:A2:9C:EA:14:0A
Signature algorithm name: SHA1withRSA
Version: 3
```

Figura 35 - Obtenció del certificat

- Un cop tinguem la clau del certificat per registrar la nostra aplicació, anirem a la web de Google Developers Console i crearem una nova aplicació.

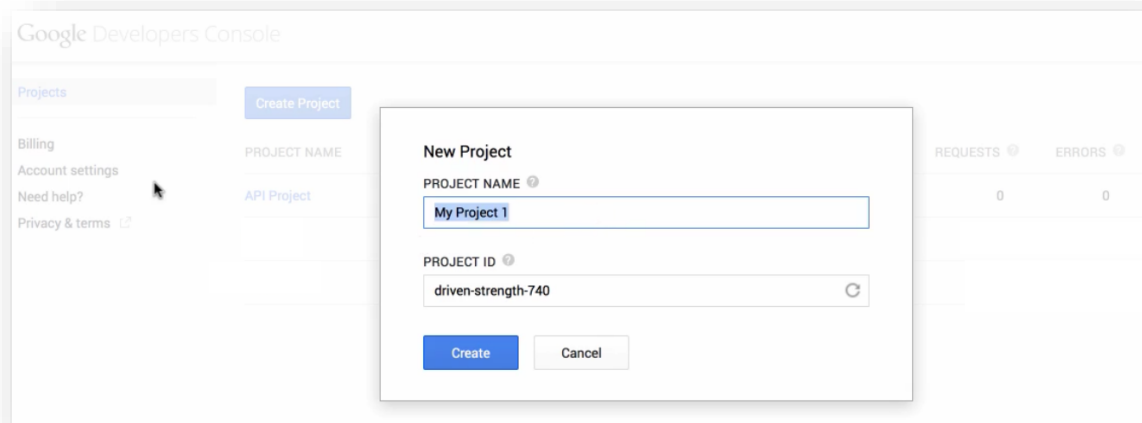


Figura 36- Creació nova aplicació Google

- Tot seguit, afegirem les API's al projecte que vulguem utilitzar, en aquest cas la de Google Maps i la de Google Places:



Figura 37 - Activació API's Google

- Després d'haver determinat quins serveis (API's) volem utilitzar, introduïrem la nostra clau del certificat signat que hem generat als primers passos, que es la clau SHA1:

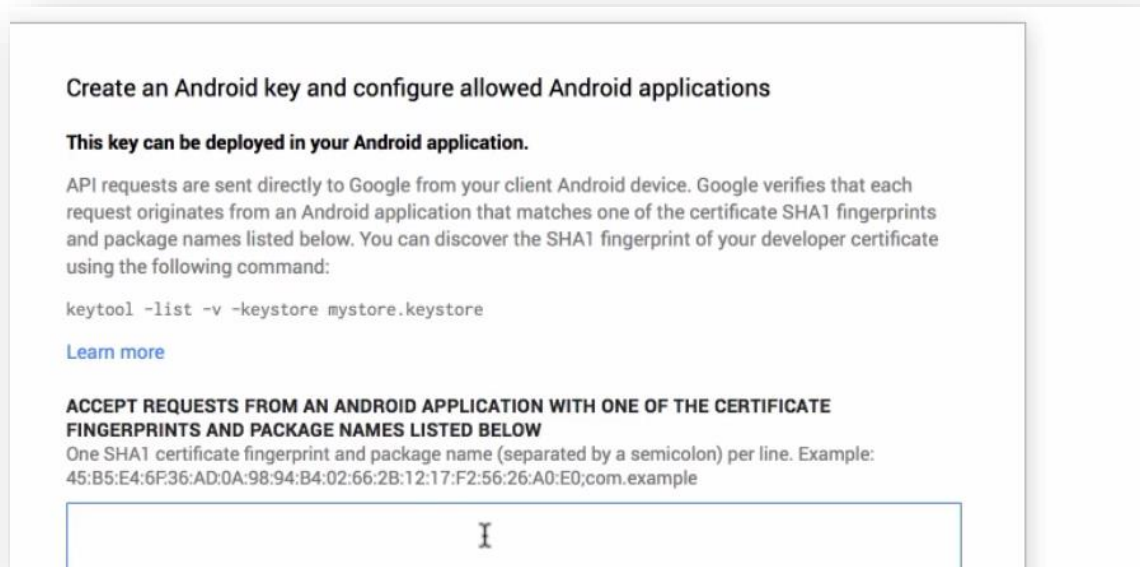


Figura 38- Activació del certificat Google

- Al introduir la nostra clau de certificat, ens assignarà una clau de aplicació, que es la que finalment tindrem que introduir al AndroidManifest.xml per tal de fer la connexió entre els serveis de Google i la nostra aplicació.



```

<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="API_KEY"/>

</application>

```

Figura 39 - Fragment de codi AndroidManifest.xml

Finalment, un cop seguits aquets passos, ja podrem utilitzar els serveis de Google en la nostra aplicació i jugar amb ells amb els components de les nostres activitats.

### Implementació Google API's (Maps + Places)

Finalitzada la integració de les API's de Google, explicarem com utilitzar els components d'aquestes API's, per tal de poder jugar amb la GEO localització i els llocs d'interès que ens proporcionen *Maps* i *Places*.

Primerament parlarem de Maps. Per tal de poder visualitzar la localització del usuari que utilitzi la nostra aplicació, tindrem que definir uns permisos de GEO localització que l'usuari tindrà que acceptar a l'hora de la instal·lació. Els permisos s'inclouran dins del *AndroidManifest.xml*.

```

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<!-- The following two permissions are not required to use
     Google Maps Android API v2, but are recommended. -->
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>

</manifest>

```

Figura 40 - Fragment de codi AndroidManifest.xml

Per altra banda, també tindrem que definir els permisos de OpenGL<sup>20</sup> per tal de poder visualitzar el nostre mapa a la nostra aplicació.

<sup>20</sup> **OpenGL**: és una especificació estàndard que defineix una API multilinguatge i multiplataforma per a escriure aplicacions que produeixen gràfics 3D.

### Specify requirement for OpenGL ES version 2

The Google Maps Android API uses OpenGL ES version 2 to render the map. If OpenGL ES version 2 is not installed, your map will not appear. We recommend that you add the following `<uses-feature>` element as a child of the `<manifest>` element in `AndroidManifest.xml`:

```
<uses-feature
    android:glEsVersion="0x00020000"
    android:required="true" />
```

Figura 41 - Codi activació OpenGL

L'últim que ens queda ara es afegir el fragment de mapa a l'activitat que nosaltres desitgem, i configurar-lo amb les coordenades i marques corresponents.

```
// inflate and return the layout
View v = inflater.inflate(R.layout.fragment_mapview, container,
    false);
mMapView = (MapView) v.findViewById(R.id.mapView);
mMapView.onCreate(savedInstanceState);

mMapView.onResume(); // needed to get the map to display immediately
try {
    MapsInitializer.initialize(getActivity().getApplicationContext());
} catch (Exception e) {
    e.printStackTrace();
}

googleMap = mMapView.getMap();
```

Figura 42- Codi exemple API mapa de Google

Després de parlar de com afegir un component *Map* a la nostra aplicació, parlarem de com afegir un component de *Google Places*.

Aquest component no serà una nova sub-activitat que serà cridada en la nostra activitat desitjada.

```

try {
    PlacePicker.IntentBuilder intentBuilder =
        new PlacePicker.IntentBuilder();
    intentBuilder.setLatLngBounds(LLEIDA_VIEW);
    Intent intent = intentBuilder.build(getApplicationContext());
    startActivityForResult(intent, PLACE_PICKER_REQUEST);

} catch (GooglePlayServicesRepairableException e) {
    e.printStackTrace();
} catch (GooglePlayServicesNotAvailableException e) {
    e.printStackTrace();
}

```

Figura 43 - Codi exemple API Google Places

Un cop creada aquesta sub-activitat, la nostra activitat creadora estarà esperant una resposta d'aquesta sub-activitat per tal de poder agafar les dades que li passin, que en aquest cas serà la GEO localització i el carrer i localització del lloc que hem triat.

```

//Get data from the activity Placer picker
if (requestCode == PLACE_PICKER_REQUEST
    && resultCode == Activity.RESULT_OK) {

    final Place place = PlacePicker.getPlace(data, this);
    final CharSequence name = place.getName();
    final CharSequence address = place.getAddress();
}

```

Figura 44 - Codi exemple retorn API Google Places

## Implementació Facebook Parse Login

La implementació de Facebook ve lligada a la API de Parse abans explicada. L'únic que tenim que fer es utilitzar la funció que ens proporciona la mateixa API, i ella sola ens genera un usuari a través del usuari que tenim identificat al nostre dispositiu, i després d'això el guarda a la nostra base de dades.

```

final List<String> permissions = Arrays.asList("public_profile", "email");
public void onLoginClick(View view)
{
    ParseFacebookUtils.logInWithReadPermissionsInBackground(this, permissions, new LogInCallback() {
        @Override
        public void done(final ParseUser user, ParseException err) {

```

Figura 45 - Codi exemple integració de Facebook Parse

El codi es breu i fàcil d'aplicar:

Utilitzem el objecte que ens proporciona la API, que en aquest cas en **ParseFacebookUtils**. Tot seguit, cridem a la funció **loginWithReadPermissionsInBackground** i li passem els permisos que volem obtenir de la conta amb la qual accedirem.

Aquesta funció funciona com les esmentades anteriorment, la aplicació crea un sub-fil en AsyncTask que s'executa en background. Així l'usuari no sofreix cap parada a l'hora de l'execució de l'aplicació.

## Implementació de diferents idiomes

Per poder implementar l'aplicació en diferents idiomes, l'únic que tenim que fer es crear diferents carpetes a l'apartat de recursos del nostre projecte, amb el nom de *values*, seguit d'un guió i el codi de idioma que volem implementar.

Aquest nou directori contindrà un arxiu igual a totes les carpetes de diferents *values-language* i segons el idioma del dispositiu, la aplicació farà servir un recurs d'idioma o un altre.

Inserirem una imatge perquè la jerarquia de directoris i valors quedi més clara:

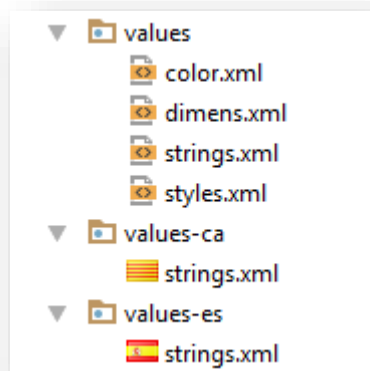


Figura 46 - Directoris diferents idiomes

## Proves de funcionament

Durant tot el desenvolupament de l'aplicació, hem tingut una fase de proves de l'aplicació, les quals, al no ser un projecte professional, hem anat duent a terme a ritme de creació de la nostra aplicació. Destacarem les més importants.

### Proves integració

Verificació que els components individuals funcionen de manera correcta. Tot seguit, es verifica que tels components junts també ho fan de manera exemplar. Les proves de integració determinen el funcionament dels components de la nostra aplicació i la integració de aquests mateixos units.

## Proves de sistema

Aquest tipus de prova, determina si l'aplicació funciona en diferents dispositius i en diferents versions de Android. La prova ha sigut satisfactòria.

També verifica que el model de dades, l'obtenció i enregistrament d'aquestes funciona correctament. A part de tot això, també comprovem que l'aplicació compleix tots els casos de ús que hem determinat, i les seves referències.

## Resultats finals

Un cop finalitzada la implementació del projecte, ens hem disposat a provar-lo i a penjar-lo a **Google Play**<sup>21</sup>, perquè la gent tingui un més fàcil accés a l'hora de baixar-ne la aplicació al seu dispositiu. El resultat ha sigut tot un èxit, ja que porta més de 20 descàrregues en els seus primers dies de vida, i a la gent li ha agradat molt el disseny de l'aplicació.

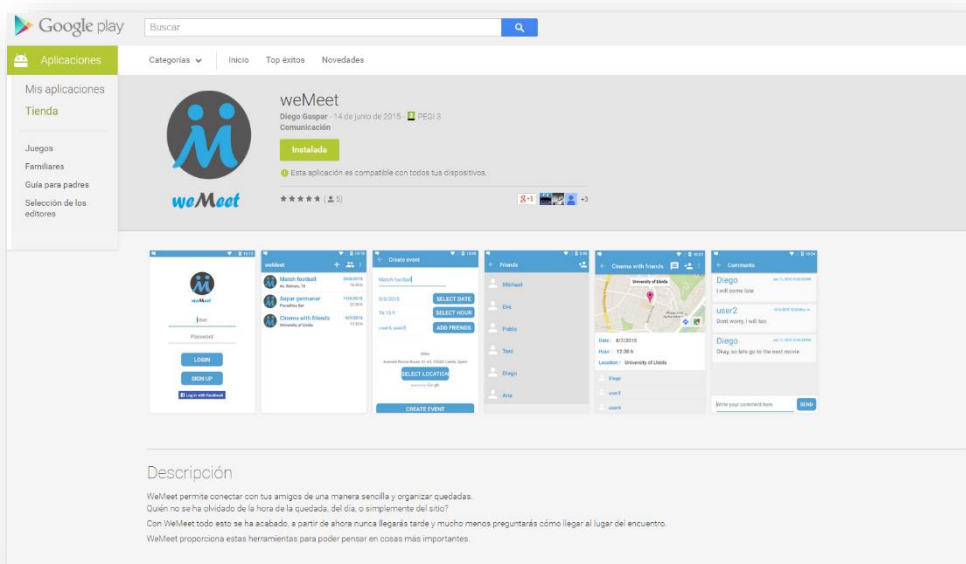


Figura 47 - Aplicació a Google Play

El que farem ara es comentar l'aplicació final, ajudant-nos de captures de pantalla fetes en un dispositiu mòbil, en aquest cas un *Nexus 4 amb versió Android 5.0 Lollipop*. Així podrem observar els resultats finals i esperats, a partir de totes les tasques que ens vam proposar, així com els casos de ús que vam determinar, i veurem les tecnologies utilitzades.

<sup>21</sup> **Google Play** : Plataforma de distribució d'aplicacions per a dispositius mòbils amb sistema operatiu Android.

## Splash, accés i creació de usuaris

Les primeres captures que presentarem seran les de les activitats principals, les quals tots els usuaris es trobaran al engegar la aplicació i no hagin accedit a l'aplicació o simplement es vulguin registrar en aquesta.

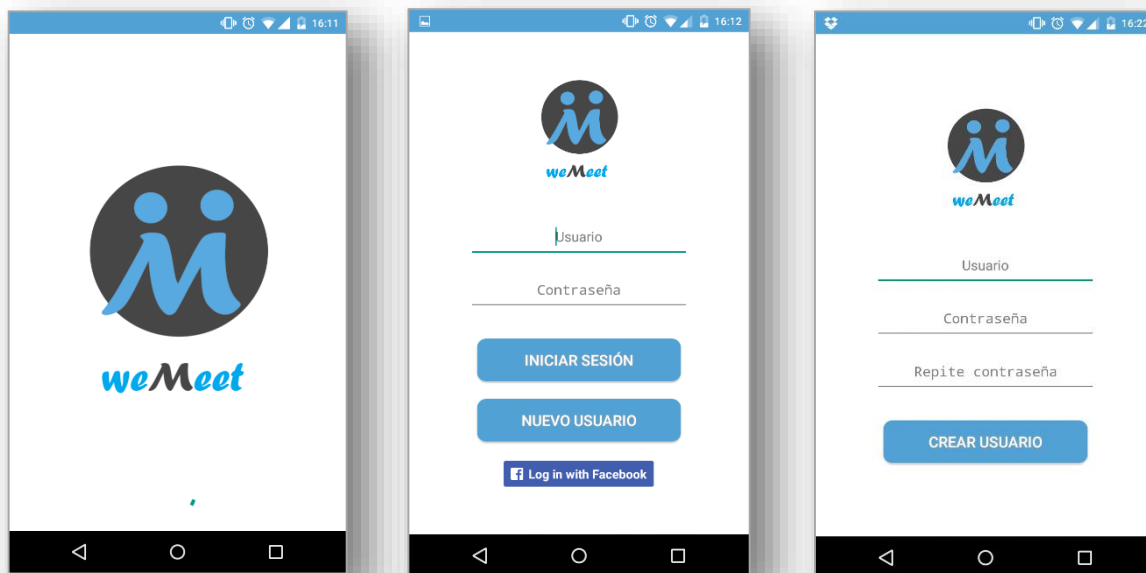


Figura 48 - Splash, accés i creació usuari

La primera captura correspon al *Splash*, que no es més que una pantalla on es mostra el logotip de l'aplicació i una rodona carregant, que es un temporitzador de 2 segons per tal de que el usuari pugui apreciar el nom de l'aplicació.

La segona captura, mostra el accés de usuaris a la nostra aplicació. Conté els formularis de usuari i contrasenya, com també les opcions de crear un nou usuari o accedir via Facebook.

La tercera captura correspon a la creació de usuari, es un simple formulari on qualsevol persona pot registrar-se en la nostra aplicació. Un cop registra't, es redirigeix automàticament a la pantalla de inici de sessió, amb els credentials de registre, per comoditat i per no repetir paràmetres de introducció de l'usuari.

## Pantalla principal

La pantalla principal mostra tots els esdeveniments els quals l'usuari que ha accedit a l'aplicació està involucrat. Per cada esdeveniment, mostra la data, hora i lloc on es realitzarà. També s'ha cregut convenient utilitzar una imatge del logotip de cada esdeveniment per de cara a un futur permetre a l'usuari canviar la imatge, segons quin esdeveniment sigui. També mostra el títol de l'esdeveniment.

A part de mostrar els esdeveniments, també tenim 3 botons d'acció a la barra de opcions de pantalla, la *ActionBar*.

- El primer d'ells, ens porta a la creació d'un nou esdeveniment.
- El segon, amb icona de usuaris, ens porta a la nostra llista de amics.

El tercer són les opcions de la activitat, i ens permet tancar la sessió de usuari.

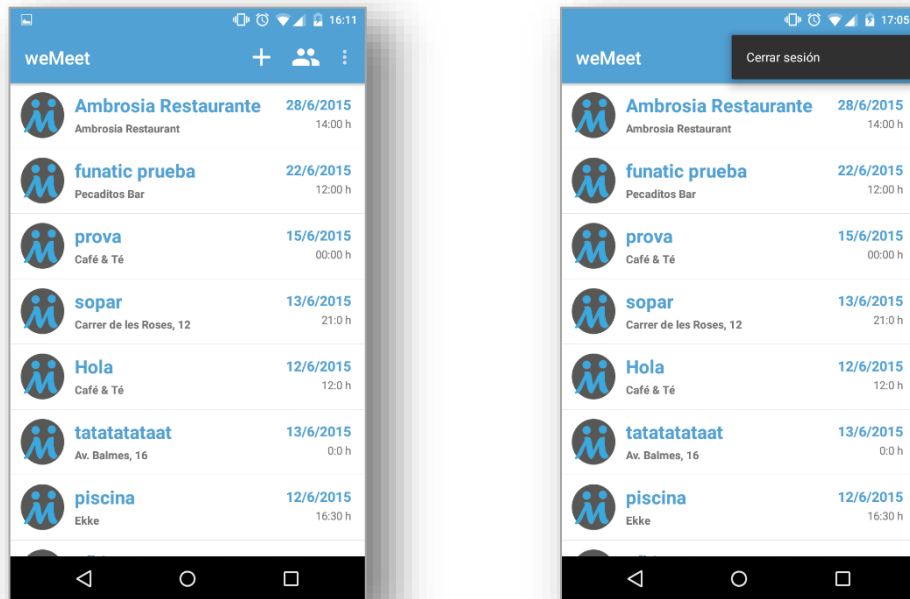


Figura 49 - Pantalla principal

## Llista d'amics

En les imatges que mostrarem a continuació, podem veure la llista d'amics que tenim afegits, com també la pantalla de opcions de afegir nous amics. Només es poden afegir amics registrats a la nostra aplicació. Un cop afegeixes a un amic, automàticament el amic afegit t'afegeix a tu també a la seva llista d'amistats.

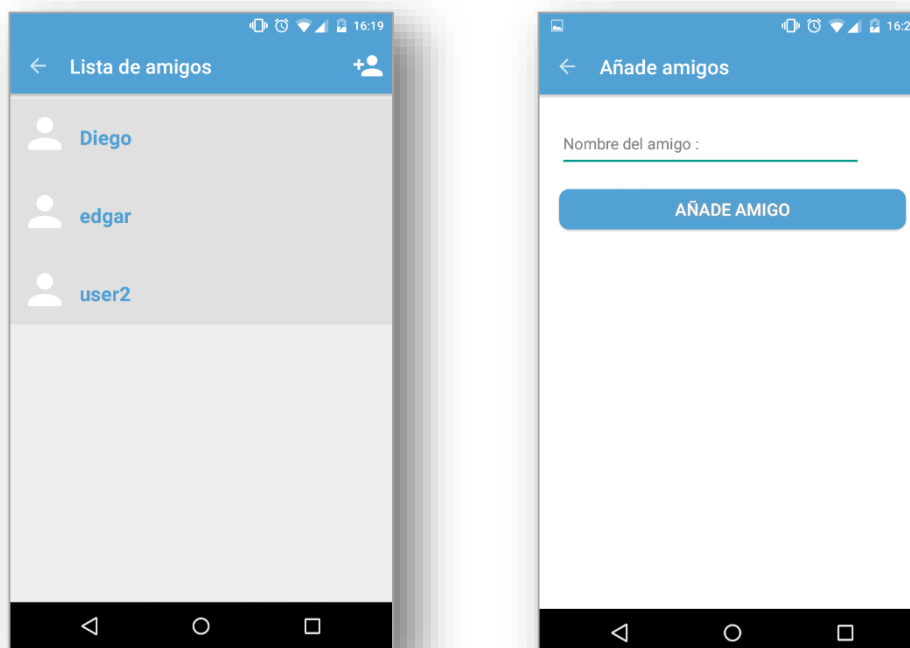


Figura 50- Opciones d'amics



- A l'esquerra, la llista d'amistats afegides.
- A la dreta, la pantalla de afegir amics a la nostra llista d'amistats.

## Creació d'esdeveniments

Quan l'usuari vol crear un esdeveniment, s'obrirà la següent pantalla mostrada a baix. També podem visualitzar com l'usuari seleccionarà la hora, així com la data, la localització i la llista d'amistats. Un cop definides totes les dades de l'esdeveniment, l'usuari podrà crear-lo.

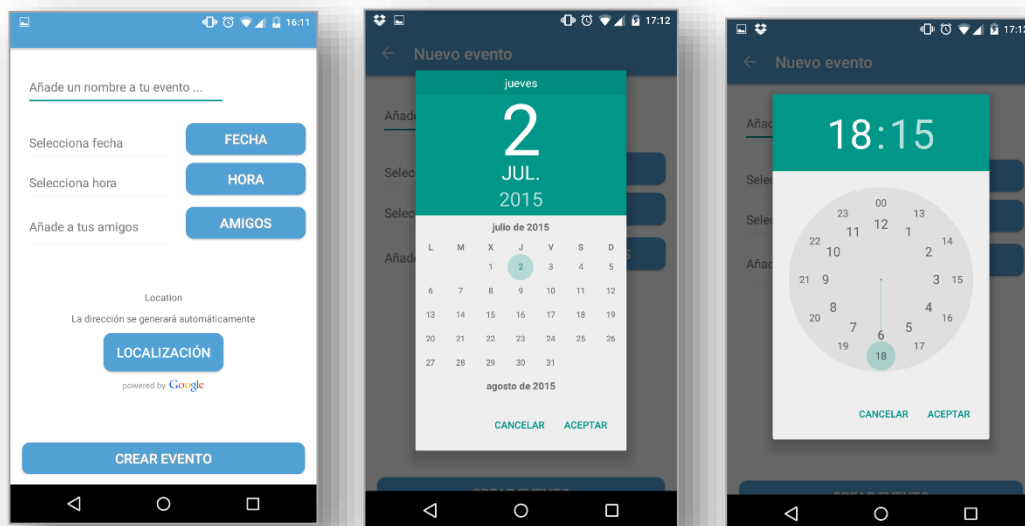


Figura 51 - Opcions creació d'esdeveniments

- Podem observar la pantalla de creació de esdeveniment, la de selecció de data en un calendari i la de selecció de hora.

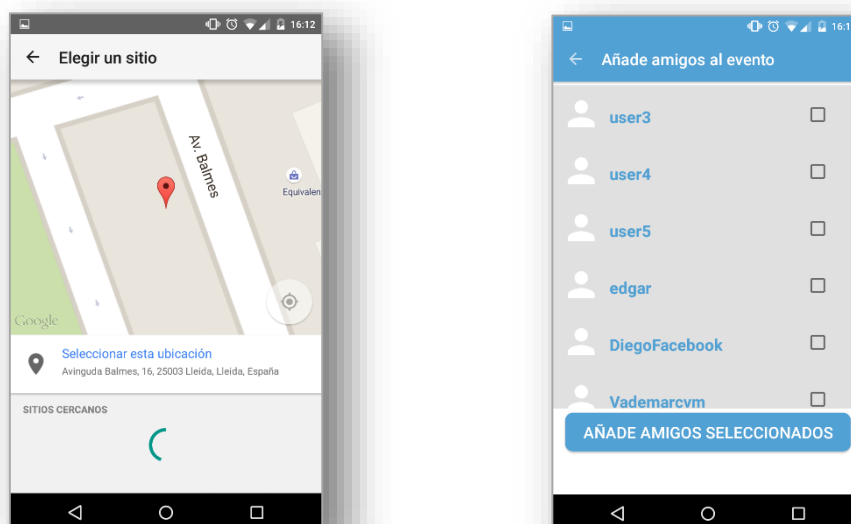


Figura 52- Opcions creació d'esdeveniments (2)

- En la primera imatge podem observar la selecció de localització i lloc de l'esdeveniment, ajudats de la API de Google Places.
- En la segona podem observar l'afegiment de amics en mode de llista i de tria a partir de caixes de selecció.

## Detall d'esdeveniment i xat

Per últim, comentarem el detall dels esdeveniments, així com la opció de xatejar dins d'aquests, entre els usuaris que estan invitats a l'esdeveniment.

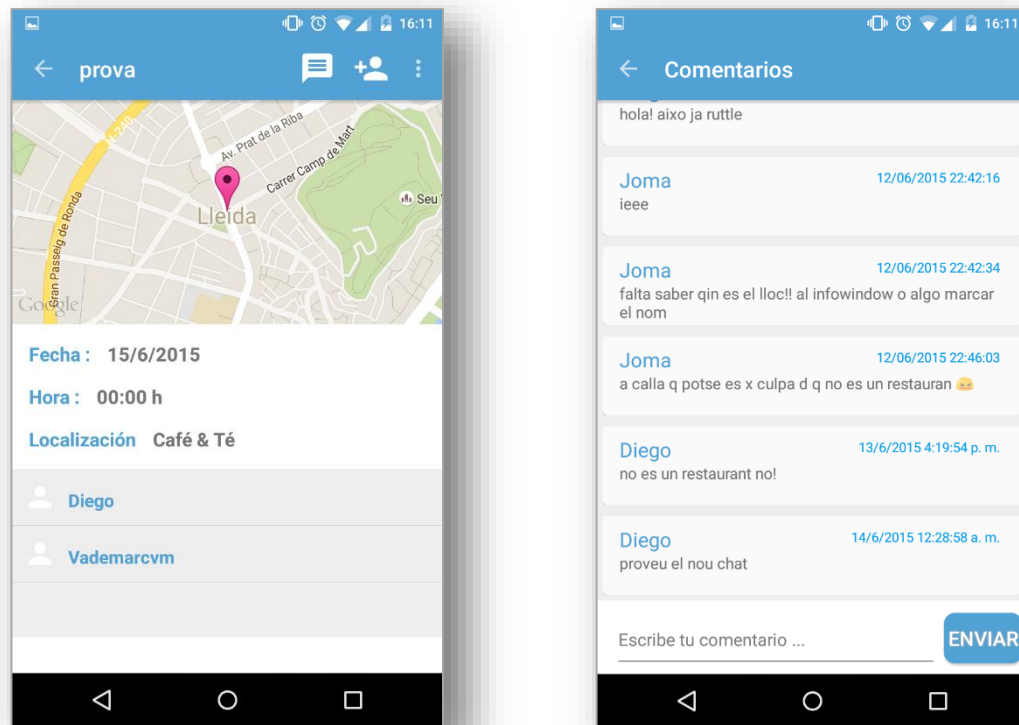


Figura 53 - Detalls esdeveniment i xat

- En la imatge de la esquerra, podem observar el detall d'esdeveniment propi, amb el mapa, indicant la localització del lloc, així com les dades de data i hora més a baix. També tenim una llista d'usuaris assistents a l'esdeveniment.
- En la segona imatge, podem observar la opció de xat d'esdeveniment. Podem interactuar amb els altres usuaris que també assistiran a l'esdeveniment.

## Conclusions

Quan parlem de les conclusions, podem afirmar que el projecte que hem plantejat s'ha assolit totalment, inclús podem dir que hem afegit millores a una idea que potser no estava clara al cent per cent des de un bon principi.

Produir i fer aquest projecte des de zero, ha fet que pugui aconseguir un coneixement important de la creació d'aplicacions mòbils per al sistema operatiu Android, i fins i tot, dedicar-me a aquest àmbit.

L'aplicació, després de ser penjada a *Google Play* i exhibida a les amistats més properes i no tant properes, he de dir que ha sigut tot un èxit. La idea en general ha estat molt adequada, encara que per suposat queden moltes coses a fer perquè puguem dir que es tracta de caràcter professional, però està clar que tot té millora i requereix molta més dedicació futura.

Cal destacar que a dies d'ara, l'aplicació ha sigut revisada per usuaris experimentats, i això ha fet que se m'obrin moltes portes futures en quant parlem de món laboral.

Com a conclusió final, he de destacar que he gaudit molt fent aquesta aplicació, i que des de un principi, volia fer una treball final de grau que hem plenes completament i no se'm fes gens pesat. Òbviament he tingut molts problemes i dies feixucs on l'únic que desitjava era trobar la solució per passar a la següent fase, però crec que en tot moment d'aprenentatge i desenvolupament aquests moments existeixen, i són els que realment et fan adquirir una bona experiència en l'àmbit de la implementació de software.

## Treball futur

Com treball futur, a l'aplicació se li podrien afegir moltes coses, però fent un breu resum, podríem dir que les més importants són les de interacció de usuaris i gestió de preferències de usuaris, així com una major facilitat de creació de esdeveniments.

Podem definir els següents punts clars, per un treball d'expansió de l'aplicació :

- **Notificacions** : Crec que és a dia d'avui una de les parts més importants de Android, ja que sense notificacions, els usuaris no entren a la teva aplicació, i s'obliden d'ella. Amb aquest aspecte, recordes a l'usuari que té nous missatges, nous esdeveniments o noves amistats a la nostra aplicació, i per tant, ha d'entrar més sovint i utilitzar-la més.
- **Modificació i gestió de perfil**: Quan parlem de modificacions de gestions de perfil parlem de que el usuari pugui jugar amb el seu perfil, canviar el seu nom, la seva contrasenya, eliminar la conta, ficar-se una foto de perfil, i tot el relacionat amb el que tots sabem de les xarxes socials. Crec que és un dels pilars de qualsevol aplicació a dies d'ara, que acompanya al sector de les notificacions que hem parlat anteriorment.
- **Facilitat creació esdeveniments**: Al llarg del "testing" entre amistats, he pogut observar que desitgen que la selecció de localització sigui a partir de un escrit de localització, i no només de marcar el punt al mapa com tenim nosaltres implementat.

És una part menys important que les altres esmentades, però tot el que tingui a veure amb l'acceptació de l'usuari i la seva fluïdesa dins de l'aplicació, s'ha de tenir sempre en compte.

# Bibliografia consultada

## Part autodidacta implementació

1. **Curs de programació en Android - Android Studio Course. Learn Android, build Apps. Lollipop** - [en línia]: <https://www.udemy.com/android-tutorial-from-beginner-to-professional/> [Consulta: Febrer i Març]
1. **Darwin, Ian F.** AndroidCookBook [en línia]: <http://androidcookbook.com/home.seam>. [Consulta: Febrer i Març]
2. **Simple Login and Signup Tutorial - AndroidBegin** [en línia] : [www.androidbegin.com/tutorial/android-parse-com-simple-login-and-signup-tutorial/](http://www.androidbegin.com/tutorial/android-parse-com-simple-login-and-signup-tutorial/) [Consulta: 25 Març]
3. **Relations Developer Guide - Parse** [en línia] : [https://parse.com/docs/relations\\_guide](https://parse.com/docs/relations_guide) [Consulta: 25 Abril]
4. **How to set date in editText - StackOverFlow** [en línia] : [stackoverflow.com/questions/11360915/how-to-set-date-in-edit-text](http://stackoverflow.com/questions/11360915/how-to-set-date-in-edit-text) [Consulta: 26 Abril]
5. **Start an activity from a fragment - Stack Overflow** [en línia] : [stackoverflow.com/questions/15478105/start-an-activity-from-a-fragment](http://stackoverflow.com/questions/15478105/start-an-activity-from-a-fragment) [Consulta: 26 Abril]
6. **Building a Flexible UI - Android Developers** [en línia] : [developer.android.com/training/basics/fragments/fragment-ui.html](http://developer.android.com/training/basics/fragments/fragment-ui.html) [Consulta: 28 Abril]
7. **Android Developer Guide - Parse Guide** [en línia] : [https://parse.com/docs/android\\_guide](https://parse.com/docs/android_guide) [Consulta: 29 Abril]
8. **Tutorial sobre la Action Bar en Android - Hermosa Programación** [en línia] : [www.hermosaprogramacion.com/2014/09/android-action-bar.html](http://www.hermosaprogramacion.com/2014/09/android-action-bar.html) [Consulta: 2 Maig]
9. **API de Google Places. - Google developers** [en línia] : <https://developers.google.com/places/documentation/?hl=es>. [Consulta: 5 Maig]
10. **Android Icons - icons4Android** [en línia] : [icons4android.com](http://icons4android.com) [Consulta: 7 Maig]
11. **Parse.com login with facebook- Stack Overflow** [en línia] : [stackoverflow.com/questions/17265379/parse-com-login-with-facebook-strange-crash-on-android](http://stackoverflow.com/questions/17265379/parse-com-login-with-facebook-strange-crash-on-android) [Consulta: 10 Maig]

12. **Facebook.** API de Facebook. [en línia] :  
<https://developers.facebook.com/docs/android> [Consulta: 12 Maig]
13. **Android TimePicker Dialog ( TimePickerDialog ) - Youtube** [en línia] :  
<https://www.youtube.com/watch?v=GDAjPcXRZho> [Consulta: 14 Maig]
14. **Google Maps Android API - Google Developers** [en línia] :  
<https://developers.google.com/maps/documentation/android/start?hl=es> [Consulta: 23 Maig]
15. **Android OS: Add GoogleMap as fragment @ Joel Lipman .Com** [en línia] :  
[www.joellipman.com/articles/google/android-o-s/app-development/745-android-os-add-googlemap-as-fragment.html](http://www.joellipman.com/articles/google/android-o-s/app-development/745-android-os-add-googlemap-as-fragment.html) [Consulta: 31 Maig]
16. **Supporting Different Languages - Android Developers** [en línia] :  
[developer.android.com/training/basics/supporting-devices/languages.html#CreateDirs](http://developer.android.com/training/basics/supporting-devices/languages.html#CreateDirs)  
[Consulta: 1 Juny]
17. **How to pass data from 2nd activity to 1st activity when pressed back? - Stack Overflow** [en línia] : [stackoverflow.com/questions/14292398/how-to-pass-data-from-2nd-activity-to-1st-activity-when-pressed-back-android](http://stackoverflow.com/questions/14292398/how-to-pass-data-from-2nd-activity-to-1st-activity-when-pressed-back-android) [Consulta: 5 Juny]

## Part informativa memòria

1. **Android - Viquipèdia, l'enciclopèdia lliure** [en línia] :  
<https://ca.wikipedia.org/wiki/Android> [Consulta: 5 Juny]
2. **los - Wikipedia, la enciclopedia libre** [en línia] : <https://es.wikipedia.org/wiki/los>  
[Consulta: 5 Juny]
3. **Android tendría una cuota de mercado del 45% en 2015 - Ocio** [en línia] :  
[www.ocio.net/tecnologia/android-tendria-uno-cuota-de-mercado-del-45-en-2015/](http://www.ocio.net/tecnologia/android-tendria-uno-cuota-de-mercado-del-45-en-2015/)  
[Consulta: 7 Juny]
4. **Anàlisi DAFO - Viquipèdia, l'enciclopèdia lliure** [en línia] :  
[https://ca.wikipedia.org/wiki/Anàlisi\\_DAFO](https://ca.wikipedia.org/wiki/Anàlisi_DAFO) [Consulta: 10 Juny]
5. **Desarrollo ágil de software - Viquipèdia, l'enciclopèdia lliure** [en línia] :  
[http://es.wikipedia.org/wiki/Desarrollo\\_%C3%A1gil\\_de\\_software](http://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software). [Consulta: 16 Juny]